3D Gaze Estimation from Remote RGB-D Sensors

THÈSE N^{O} 6680 (2015)

PRÉSENTÉE LE 9 OCTOBRE 2015

À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR

LABORATOIRE DE L'IDIAP

PROGRAMME DOCTORAL EN GÉNIE ÉLECTRIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Kenneth Alberto FUNES MORA

acceptée sur proposition du jury:

Prof. K. Aminian, président du jury
Dr J.-M. Odobez, directeur de thèse
Prof. L.-Ph. Morency, rapporteur
Prof. D. Witzner Hansen, rapporteur
Dr R. Boulic, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2015

To my family and friends...

Acknowledgements

This PhD has been one of the most amazing chapters in my life. During these highly dynamic and intense years one element remained constant: I was always lucky to be surrounded by incredible people who were always willing to discuss, to share achievements, to exchange ideas, to party or simply to hang out, and that never hesitated to share their knowledge or to provide advice in times of need. Here, I would like to acknowledge them as it is due.

First of all, I would like to thank Jean-Marc, my supervisor. Your expertise and guidance, always insightful, meticulous and timely, was crucial for the success of this thesis. Moreover, I will always be thankful for your constant dedication, many advices, trust and support, which went beyond the PhD work, and which will have a long lasting impact in my future endeavours. I also want to thank Prof. Kamiar Aminian, Dr. Ronan Boulic, Prof. Louis-Philippe Morency and Prof. Dan Witzner Hansen for kindly taking part of my thesis jury.

I want to thank the Swiss National Science Foundation for providing the financial support that made this PhD possible. During these years, I had the opportunity to collaborate directly with great people: Laurent, Daniel, Florent, Matthieu, Navid, Samira, Yann, François, Catha and Jocke. Thank you for your work and scientific input that helped to enrich this thesis.

It was an honour to be part of the Perception and Activity Understanding group, to exchange ideas, knowledge, experiences, tips and know-hows. Thanks to all of you, the current members: Alex, Rui, Gülcan, Paul, Nam, Wu Di and Yiqiang, as well as the former members I had the honour to meet: Samira, Rémi, Adolfo, CC, Jagan, Vasil, Stefan, Carl, Romain, Navid and Matthieu.

Idiap was a great place where I was able to work with bright people, but it also gave me the opportunity to gain valuable friends. Alex and Samira, your awesomeness is limitless, I shared great times with you and I hold dearly our friendship. The same goes for Adolfo and the many times we rocked, as well as Rémi and Majid, the cool guys and good friends, crazy enough to have joined me into ICC. Thanks to the people from 308, the best office ever (period), and all the cool people around: Alex, Cijo, Rui, Gülcan, Marc, Vicky, Ivana, Raphael, Radu, Kate, Maryam, Nam, Wu Di, Dinesh, Nikos, Sharid, Joan, Marco, Manuel, Thomas, Leo, Serena, Nesli, Ufuk, Phil, Tiago, Pierre-Edouard, James, Laurent N. and Laurent E., Elie, Claire, Charles, Mihn Tri, Petr, Dayra, Xiao, Laskmi, André, Marzieh, Novi, Ramya, Afsaneh, Harsha, Pranay, Tatjana, Aleksandra, Alexandros, Paco, and many others. I also want to thank Nadine and Sylvie, for

always being so nice and helpful with every little detail we needed. Thanks to everyone from the administrative, direction and system staff, always kindly addressing our requests and keeping Idiap running. At EPFL, I want to thank Corinne, Chantal and Vanessa for their work during my thesis organisation, as well as Prof. Jean-Philippe Thiran and Frank, Nicolas and Arvind, with whom I enjoyed working in the EDEE Student Committee.

I want to thank the tunabrix band: Carlos, Pol, Dimitri, Róger, Mirjam and Mario, as well as the great friends from Lausanne: Emilie, Amiel, Agnieszka, Ele, Parag and Mandana, and my former flatmates Malik and Danielle. All of you made daily life much more interesting and contributed to be able to make it through the ups and downs of the PhD.

During these years, I had the blessing of counting with great friends. Konstantin, mate, I want to thank you for being such an amazing and unfailing friend that I could always count on. Thanks also to my old but still very close friends: Lisseth, Ernesto, Lucia, Luis, Tavo, Sterling, Jorge, Randall and Esthepanie, who have always been there, in a way or the other, despite time and distance. Thanks also to all the people involved in the Vibot program, which made it possible for me to be here in the first place. In addition, I want to thank Dr. Pablo Alvarado, for his early influence and support, as well as to all my former TEC and Canam colleagues.

I want to thank Catha, for your always positive attitude, great friendship, constant support and encouragement. I look forward to many new adventures together, sweetie. Finally, I want to thank my parents, Carlos and Ruth, for their love, care and help, despite the long distances and the many challenges we have faced. All this has been possible thanks to you and for you.

Lausanne, 12 July 2015 Kenneth A. Funes Mora

Abstract

The development of systems able to retrieve and characterize the state of humans is important for many applications and fields of study. In particular, as a display of attention and interest, gaze is a fundamental cue in understanding people activities, behaviors, intentions, state of mind and personality. Moreover, gaze plays a major role in the communication process, like for showing attention to the speaker, indicating who is addressed or averting gaze to keep the floor.

Therefore, many applications within the fields of human-human, human-robot and human-computer interaction could benefit from gaze sensing. However, despite significant advances during more than three decades of research, current gaze estimation technologies can not address the conditions often required within these fields, such as remote sensing, unconstrained user movements and minimum user calibration. Furthermore, to reduce cost, it is preferable to rely on consumer sensors, but this usually leads to low resolution and low contrast images that current techniques can hardly cope with.

In this thesis we investigate the problem of automatic gaze estimation under head pose variations, low resolution sensing and different levels of user calibration, including the uncalibrated case. We propose to build a non-intrusive gaze estimation system based on remote consumer RGB-D sensors. In this context, we propose algorithmic solutions which overcome many of the limitations of previous systems. We thus address the main aspects of this problem: 3D head pose tracking, 3D gaze estimation, and gaze based application modeling.

First, we develop an accurate model-based 3D head pose tracking system which adapts to the participant without requiring explicit actions. Second, to achieve a head pose invariant gaze estimation, we propose a method to correct the eye image appearance variations due to head pose. We then investigate on two different methodologies to infer the 3D gaze direction. The first one builds upon machine learning regression techniques. In this context, we propose strategies to improve their generalization, in particular, to handle different people. The second methodology is a new paradigm we propose and call geometric generative gaze estimation. This novel approach combines the benefits of geometric eye modeling (normally restricted to high resolution images due to the difficulty of feature extraction) with a stochastic segmentation process (adapted to low-resolution) within a Bayesian model allowing the decoupling of user specific geometry and session specific appearance parameters, along with the introduction of priors, which are appropriate for adaptation relying on small amounts of data. The aforementioned gaze estimation methods are validated through extensive experiments in a

comprehensive database which we collected and made publicly available.

Finally, we study the problem of automatic gaze coding in natural dyadic and group human interactions. The system builds upon the thesis contributions to handle unconstrained head movements and the lack of user calibration. It further exploits the 3D tracking of participants and their gaze to conduct a 3D geometric analysis within a multi-camera setup. Experiments on real and natural interactions demonstrate the system is highly accuracy.

Overall, the methods developed in this dissertation are suitable for many applications, involving large diversity in terms of setup configuration, user calibration and mobility.

Keywords: gaze estimation, head pose tracking, gaze coding, RGB-D sensors, 3D, generative models, human human interaction, human robot interaction, human computer interaction.

Résumé

Le développement de systèmes capables d'estimer et de caractériser l'état des humains est important pour de nombreux champs d'étude et applications. En particulier, le regard, en tant que manifestation d'attention et d'intérêt, fournit une information primordiale pour la compréhension des activités, comportements, intentions, états d'esprit des personnes, et même de leur personnalité. Plus précisément, le regard joue un rôle majeur dans le processus de communication, par exemple pour montrer son attention à son interlocuteur, indiquer à qui l'on s'adresse, ou détourner le regard pour garder la parole.

Par conséquent, beaucoup d'applications dans les domaines des interactions humain-humain, humain-robot ou humain-ordinateur pourraient potentiellement bénéficier de la perception du regard. Cependant, malgré des avancées significatives, fruit d'une trentaine d'années de recherche, les méthodes actuelles d'estimation du regard sont incapables de traiter beaucoup de scénarios fréquents dans les domaines sus-cités, qui requièrent souvent la perception à distance, la gestion des mouvements non contraints des utilisateurs et une intervention minimale de leur part pour la phase de calibration. De plus, afin de réduire les coûts, il est préférable d'avoir recours à des capteurs issus du marché des consommateurs, qui produisent habituellement des images à basse résolution et faiblement contrastées que les systèmes actuels peinent à gérer.

Dans cette thèse, nous étudions le problème de l'estimation automatique du regard sujet à des variations de pose de la tête, à basse résolution et pour différents niveaux de calibration, incluant les scénarios sans calibration. Nous proposons la construction d'un système non intrusif d'estimation du regard utilisant des capteurs RGB-D à distance, issus du marché des consommateurs. Dans ce contexte, nous proposons des solutions algorithmiques qui surmontent beaucoup des limitations des systèmes existants. Nous traitons ainsi des aspects suivants du problème dans cette thèse : suivi de la pose de la tête en 3D, estimation du regard en 3D, et modélisation d'applications basées sur le regard.

Dans un premier temps, nous développons un modèle de suivi de la pose de la tête en 3D avec une précision élevée, qui s'adapte à l'utilisateur courant sans demander d'action explicite de sa part. Dans un second temps, de façon à s'affranchir de la pose de la tête, nous proposons une méthode de correction des variations de l'apparence de l'image de l'œil dues à la pose. Nous étudions ensuite deux méthodologies différentes pour déduire la direction du regard en 3D à partir de ces images. La première méthodologie part de techniques de régression basées sur des modèles d'apparence. Dans ce contexte, nous proposons des stratégies pour améliorer

leur généralisation, en particulier pour gérer les variations d'apparence entre utilisateurs. La seconde méthodologie est un nouveau paradigme pour l'estimation du regard appelé estimation générative géométrique du regard. Cette nouvelle approche combine les bénéfices de la modélisation géométrique de l'œil (normalement réservée aux images haute résolution en raison de la difficulté à extraire des caractéristiques) avec un processus stochastique de segmentation (adapté à la basse résolution), dans le cadre d'un modèle bayésien qui permet le découplage de la géométrie spécifique à l'utilisateur et des paramètres d'apparence spécifiques à une session, ainsi que l'introduction de connaissances a priori, utiles à l'adaptation avec peu de données. Les méthodes d'estimation du regard susmentionnées sont validées au travers d'expériences approfondies sur une base de données exhaustive que nous avons collectée et rendue publique.

Enfin, nous étudions le problème du codage automatique du regard dans des interactions naturelles entre deux ou plusieurs personnes. Le système exploite les contributions de cette thèse pour gérer les mouvements non contraints de la tête et l'absence de calibration de la part des utilisateurs. Par ailleurs, le système utilise le suivi 3D des participants et de leur regard pour conduire une analyse géométrique en 3D dans une installation multi-caméra. Des expériences sur des interactions réelles et naturelles démontrent que le système atteint une haute précision dans le codage du regard.

De manière générale, les méthodes développées dans cette thèse sont adaptées à un large éventail d'applications impliquant des configurations diverses d'installation, de calibration et de mobilité des utilisateurs.

Mots-clefs : estimation du regard, suivi de la pose de la tête, codage du regard, capteurs RGB-D, 3D, modèles génératifs, interaction humain-humain, interaction humain-robot, interaction humain-ordinateur.

Contents

List of Figures

List of Tables

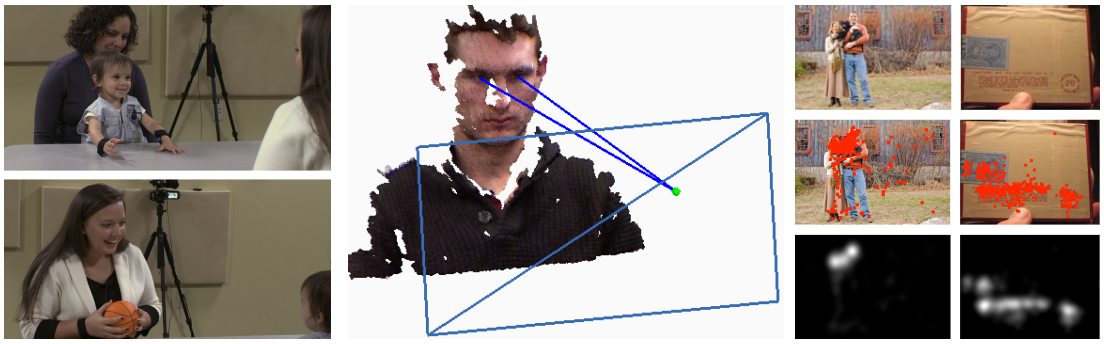
1 Introduction

The ability to effortless read and assess the physical state of other people is an innate skill for humans. We constantly monitor each other's head and body pose, our hands position and gestures, facial expressions and gaze. These elements provide rich information about people's intentions, state of mind and help clarify the spoken message. These cues, which constitute *non verbal behavior*, are therefore key elements for the perception in human interactions.

Gaze in particular is acknowledged as one of the most important non verbal communication cues. It plays a crucial role when people interact, as it is used to regulate the flow of communication, monitor feedback, reflect cognitive activity, express emotions, and communicate the nature of the interpersonal relationship [Kendon, 1967, Duncan, 1972, Cassell, 2000, Knapp and Hall, 2009]. Gaze patterns vary enormously according to the social setting. The interlocutors' personality, the conversation topic, or the other person's gaze behavior are all factors which might influence one's gaze. In non-communicative settings, gaze has been shown to be informative of underlying cognitive processes. Gaze patterns, e.g., the *gaze bias* phenomenon, are reflective of the decision making process, whether on deciding which object to select among a set [Shimojo et al., 2003, Schotter et al., 2010], or what direction to take [Wiener et al., 2012]. In general, gaze is a strong indicator of a subject's attention, and its analysis have been key to understand the human attention process and to build computational models of visual saliency and attention [Frintrop et al., 2010, Borji and Itti, 2013, Zhao and Koch, 2013].

Due to the rich information which can be extracted from non verbal cues, in recent years there has been a growing interest from diverse domains on tools able to automatically retrieve and characterize the state of humans. Gaze in particular is of high value for a wide range of fields of study and potential applications. However, despite significant advances during more than 30 years of research, many scenarios can not be addressed by current gaze estimation systems.

This thesis aims at developing a gaze estimation system which can further extend the set of plausible applications by addressing key challenges of the gaze estimation problem, such as minimal user calibration, head pose variations, remote sensing and low resolution imaging.



(a) (b) (c)

(d) (e)

Figure 1.1: Examples of applications which can profit from gaze estimation. (a) Autism diagnosis in small children: Multimodal Dyadic Behavior (MMDB) dataset [Rehg et al., 2013]. (b) Monitoring the point of fixation in a screen. Used as input or to analyze gaze patterns. (c) Gaze tracking results from a plurality of subjects, used to obtain an image visual saliency map [Bylinskii et al.]. (d) Studying Human-Human Interactions in group interviews [Oertel et al., 2014]. (e) Human-Robot Interactions involving groups of people [Jayagopi et al., 2013].

In the rest of this introductory chapter we will present relevant applications to further motivate the need for gaze sensing. Then, the problem of automatic gaze estimation will be formally defined along with its challenges. We will then state the main goal of this thesis and list its contributions. Finally, an overview of the organization of the thesis will be presented.

1.1 Motivation

As a display of attention and interest, gaze is a fundamental cue in understanding people's activities, behaviors, and state of mind. Among the many applications and fields of study which can profit from automatic gaze estimation systems are:

1. **Psychology and Sociology**. Gaze conveys information about a subject's state of mind and personality. Researchers have therefore used gaze information as a nonverbal cue to analyze face-to-face interactions and to automatically detect social variables such as

dominance [Hung et al., 2008], personality traits [Lepri et al., 2010] and group dynamics [Gatica-Perez et al., 2005, Oertel and Salvi, 2013]. The monitoring of gaze patterns may also facilitate the diagnosis of certain conditions, for example, autism in young children (see Fig. 1.1a), which tend to exhibit distinctive gaze behaviors within their social interactions [Wetherby et al., 2004]. In general, automatic gaze estimation can be highly valuable to conduct large scale studies in psychology and sociology, which have so far been restricted to manual and/or crude annotations [Gorga and Otsuka, 2010, Rehg et al., 2013]. Nevertheless, it becomes crucial to minimize the level of intrusion, as otherwise the participant's behavior may be compromised by the setup. Thus, there is a clear need for remote gaze estimation systems which do not require the user to undergo explicit calibration sessions or that impose important limits in their mobility.

2. **Human Computer and Human Robot Interaction**. There are several ways in which gaze estimation can be valuable for the development of future generations of robots, computers, virtual agents and avatars. These systems need to understand social interactions and their dynamics [Bickmore and Picard, 2005]. Using an automatic gaze estimation system, it becomes possible to collect and effectively analyze much larger corpora of human interactions from which computational models of social behavior can be derived (see Fig. 1.1d). Ultimately these computational models can be implemented into *Embodied Conversational Agents* (ECA) [Cassell, 2000, Cassell et al., 2001], and therefore improve the interaction between humans and, for example, robots (cf. Fig. 1.1e) [Mutlu et al., 2006]. Notice that effective implementations of these systems also require the real-time monitoring of non verbal behavior, inc. gaze.

3. **Computer Interfaces**. Gaze is an effective computer interface input. By monitoring where an individual is looking on a screen (cf. Fig. 1.1b), it is possible for computer programs and the OS to exploit this information. For example, the user's gaze can be used for eye-typing or to control a mouse cursor. These mechanisms are key to the development of assistive technologies for people with limited body mobility [Majaranta et al., 2011]. Overall, *gaze contingent* software may be developed for an array of applications ranging from web browsing [Rozado et al., 2015] to gaming [Isokoski et al., 2009].

4. **Cognitive Science**. The behavior of gaze is known to be reflective of underlying cognitive processes, e.g., when making decisions [Shimojo et al., 2003, Schotter et al., 2010, Wiener et al., 2012] or when it is decided to what elements and in which order to pay attention in an image (cf. Fig. 1.1c). It is therefore of interest for cognitive scientists to monitor gaze as this can shed light onto how the brain execute diverse tasks. This is also of interest to the computer vision community, as visual saliency algorithms can be learned to help machines mimic this behavior [Bylinskii et al., Zhao and Koch, 2013].

5. **Automotive Industry**. By automatically monitoring the driver's attention it is possible to design valuable safety mechanisms, e.g., it can be inferred whether the driver has seen traffic signs, pedestrians or other vehicles [Ishikawa et al., 2004]. Furthermore,

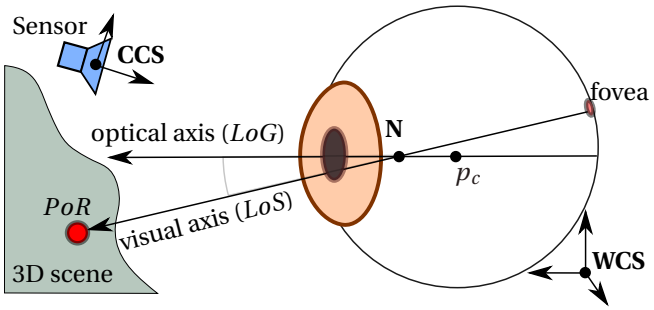


Figure 1.2: 3D gaze estimation problem. Showing a simplified diagram of the human eyeball positioned with respect to the World Coordinate System (**WCS**). The eyeball is sensed using a remote camera, which defines the camera coordinate system (**CCS**).

the eye movements are indicatives of fatigue, drowsiness amongst and other factors [Fletcher and Zelinsky, 2009, Duchowski, 2007].

6. **Marketing Research**. Gaze estimation can be used to monitor people's reaction to different products. This can be done before it is made available to the consumer, or when it is displayed next to competitor products in a retail environment. For store owners it can bring information to optimize the layout of the different products.

7. **Web, Software and OS design**. By monitoring the way people use any of these elements, include where they gaze, to what elements they pay attention to or to which elements are not observed, it is possible to properly evaluate the effectiveness of a design, leading to further improvements.

1.2 Problem Definition and Challenges

Gaze estimation may actually refer to one of three closely related goals, depending on the context, methodology and literature. The first and more general one is the determination of the 3D *Line of Sight* (*LoS*) with respect to the World Coordinate System (**WCS**). The *LoS* is the ray pointing out from the fovea and passing through the corneal nodal point **N**, cf. Fig. 1.2 [Hansen and Ji, 2010]. As the fovea is the region of highest visual acuity in the retina, from a physiological point view, the *LoS* is the direction which provides the highest quality sensing of the object of interest when said object is projected onto the retina. The *LoS* differs from the *Line of Gaze* (*LoG*), which is the ray pointing out from the eyeball rotation center p_c through the pupil center. Notice that, within the eyeball coordinate system, the *LoG* may also be referred as the *optical axis*, whereas the *LoS* is also known as the *visual axis* (see Fig. 1.2).

The second goal is the determination of the *Point of Regard* (*PoR*), which is the point within the 3D scene at which the *LoS* is directed to. Its position can be computed simply as the intersection between the *LoS* and the 3D objects in the scene. However, due to the difficulties to characterize these objects or to retrieve the *LoS*, an alternative is to circumvent these elements and to determine the *PoR* directly. In this manner, the *PoR* can even be defined

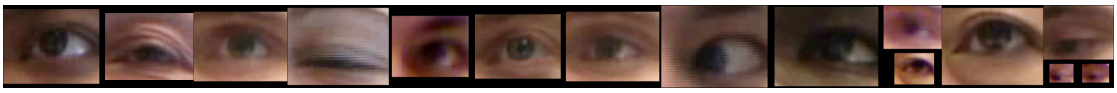


Figure 1.3: Examples of eye image variations due to user, resolution and distance to the sensor, illumination, viewpoint, head pose, etc. Images extracted from the EYEDIAP database [Funes Mora et al., 2014a].

in terms of other quantities, e.g., screen 2D pixel coordinates, or the 2D coordinates of the fixation point referred to an associated egocentric video.

Finally, the third goal refers to determining the discrete object at which the LoS is directed to; this is a definition which goes a level higher from the estimation of the PoR. Nevertheless, in this thesis we treat this problem separately and denote it as determining the *visual focus of attention* or to do *automatic gaze coding*.

Significant efforts have already been devoted to the design of automatic gaze estimation solutions, leading to methods varying according to their sensing technique and principles: from the highly intrusive electro-oculography [Hyoki et al., 1998] to the more flexible video-oculography, i.e., gaze tracking based on video input [Hansen and Ji, 2010].

Even though video-oculography has higher potential for practical applications as it is, in principle, non invasive, it needs to address important challenges. In particular, the obtained eye images vary according to the user, head pose or viewpoint, illumination conditions, image resolution or eye distance to the sensor, contrast, eyelids shape and movements, specular reflections, motion blur, self occlusions, etc., as shown in Fig 1.3. In addition, as there is direct link between head orientation and gaze [Langton et al., 2004], head pose estimation is normally required by these systems to disambiguate the gaze direction. However, head pose estimation itself is a challenging and ongoing research problem [Murphy-Chutorian and Trivedi, 2008a] whose estimation errors introduce noise into the gaze estimation algorithm.

To overcome some of these challenges, many gaze estimation systems, in particular those available in the market, rely on specialized hardware like head mounted cameras and/or infrared (IR) setups. The advantage of the former is the capture of standardized eye images, i.e., with a single scale and viewpoint and thus removing the need for head pose estimation and, due to the sensors' close distance, the eye image is normally of high resolution. Nevertheless, for many applications, head mounted sensors are considered as intrusive (see Sec. 1.1). Infrared setups (illumination and sensing) profit from the high contrast images obtained due to the bright/dark pupil effect and the reflections of calibrated IR light sources in the cornea, called *glints*. Depending on the setup configuration, these systems may accommodate head pose variations [Guestrin and Eizenman, 2006]. However, high resolution IR imaging is required, and these systems may exhibit an operation range which is very limiting for some applications.

Natural light based methods remain as the best candidates in terms of hardware availability, cost and potential applications. Yet, many of the aforementioned challenges are far from

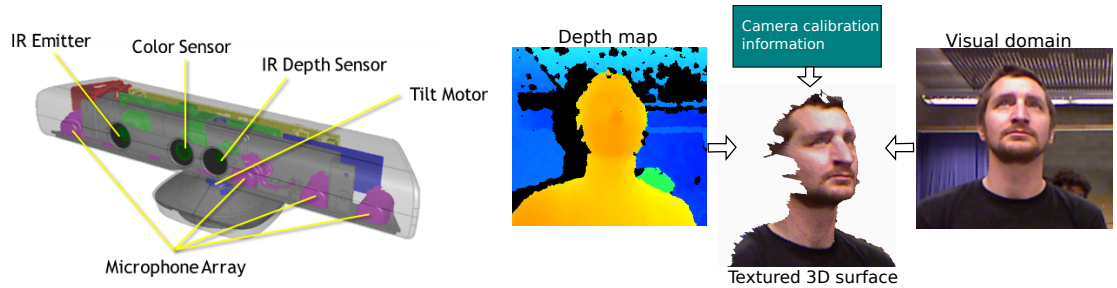


Figure 1.4: RGB-D imaging. (a) First generation Microsoft Kinect™sensor. Notice the different components, in particular the color sensor (standard camera) which forms a stereo ensemble with the depth sensor. The IR Emitter projects light patterns towards the scene, which are captured by the "IR Depth sensor" camera and used to infer depth information based on triangulation. (b) Provided the camera calibration parameters of the color/depth stereo ensemble, it is possible to combine both sources into a textured 3D mesh.

being solved. Furthermore, among the proposed natural light based gaze estimation systems, many require either high resolution imaging to track local eye features, or a cumbersome gaze calibration procedure which restricts their usage to a single session, user and, possibly, to well controlled laboratory conditions.

Although prior solutions are indeed suitable for certain scenarios, e.g., when interacting with a computer, where the objective is to retrieve the PoR as the fixation point within a flat screen, a large number of interesting scenarios and applications can not be well addressed by these systems. Examples of such scenarios are shown in Figures 1.1a, 1.1d and 1.1e. In general, these settings are challenging due to the following reasons:

- **Unintrusive gaze sensing is required**. The usage of remote sensors become mandatory, as opposed to using head-mounted systems.
- **User explicit gaze calibration may not be assumed**. It is desired or expected for the subject's behavior to be natural. Gaze calibration procedures, requiring explicit actions from the subject, may compromise his/her behavior. Moreover, such procedures are not available in applications requiring gaze sensing in the wild.
- **The head pose is unconstrained within the 3D space**. The system needs to therefore cope with eye image variations due to this factor.
- **The participants and objects of interest are defined and can move within the 3D space**. Therefore, a 3D reasoning of the problem is needed, in particular, the actual LoS needs to be determined. In addition, a sensor with a large enough field of view is required to accommodate these movements.
- **Consumer sensors are preferred or required**. This point, in combination with the need for a large field of view means that low resolution sensing of the eye region is expected.

The advent of consumer RGB-D sensors (color and depth), pioneered by the Microsoft Kinect™sensor (cf. Fig. 1.4a), may however help to address these challenges. Indeed, in

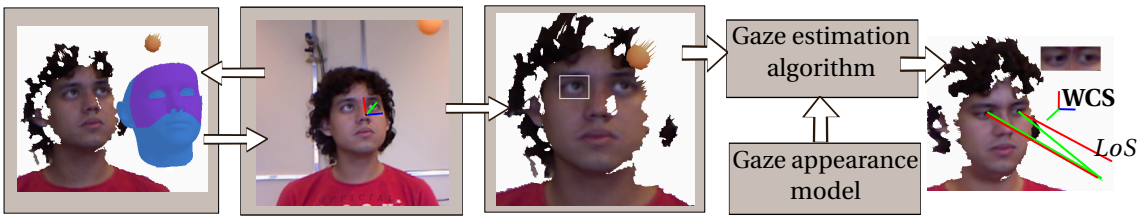


Figure 1.5: General remote 3D gaze estimation processing pipeline. (a) Person specific face model learning. (b) 3D head pose tracking. (c) Eye localization and eye image extraction. (d) Gaze estimation from the eye image appearance. (e) System output: *LoS* referred to the **WCS**.

the recent years, such sensors have allowed researchers to handle problems known to be highly challenging when based on standard vision alone, such as body pose estimation [Shotton et al., 2011] or facial expressions recognition [Weise et al., 2011]. Through *depth maps*, which are images where each pixel contains the depth distance of the object from the camera, these sensors provide explicit and reliable measurements of the scene's shape, as opposed to the implicit shape information embedded within the RGB data. This is valuable as it is still difficult and costly to infer shape information from the visual domain (RGB) alone [Barron and Malik, 2013].

Therefore, depth sensing enables the use of shape information in further processing stages. In particular, depth data has been shown to be valuable for accurate head pose estimation [Fanelli et al., 2011, Weise et al., 2011, Baltrusaitis et al., 2012], a necessary step prior to determining the gaze direction. On the other hand, gaze itself requires standard vision measurements to determine the eye orientation from the eye image, and the most important challenges to address are the eye appearance variabilities due to head pose, users, and the low eye image resolution when considering applications that do not restrict the mobility of users.

1.3 Objective and Thesis Contributions

It is now possible to summarize the main goal of this thesis as follows:

Main Objective: "*To design gaze estimation algorithms able to retrieve the 3D line of sight under unconstrained head motion and minimal user cooperation from remote consumer sensors, and to validate their usage into challenging scenarios and applications*"

User cooperation is here understood as explicit strategies required to facilitate the gaze estimation task, such as using head mounted hardware, asking the user to gaze at a set of points prior to the estimation (gaze calibration), or to maintain a particular head pose, to mention a few. For this reason, remote sensors (not head mounted) is a requirement.

Overall approach. A diagram of a general remote non intrusive gaze estimation system is shown in in Fig. 1.5. In this thesis a model based approach will be used for the head pose tracking. Therefore, the first step is the learning of a person specific face model, as shown in

Fig. 1.5a. Provided this model, the head pose can be determined by comparing the model to the data at hand (Fig. 1.5b). Once the head pose has been determined, the eye position and eye image can be extracted from the head pose parameters and the data (Fig. 1.5c). Notice in this framework the head pose tracking implicitly solves the frame by frame eye localization problem. This eye image can then be compared to a gaze appearance model using a gaze estimation algorithm, as depicted in Fig. 1.5d. Finally, given the previous estimates, it is then possible to define the *LoS* with respect to the **WCS**, as shown in Fig. 1.5e. The *LoS* estimate can then be used in further processing stages to determine the *PoR* or visual focus of attention.

1.3.1 Contributions

The contributions of this thesis are the following:

- **Depth based accurate 3D head pose tracking**. As accurate 3D head pose tracking is a requirement for remote appearance based gaze estimation systems, a framework has been proposed for 3D head pose tracking based on a face model registration to depth data. This framework builds over statistical models of shape variations: 3D Morphable Models (3DMM), to cope with facial shape differences across people. These models can span a large diversity of subjects, reduce the space of plausible facial shapes and maintain semantic consistency between instances. Two strategies have been investigated: the first one relies on the offline fitting of the 3DMM. The offline fitted model is then used for online head pose tracking. The second strategy proposes to track the head pose and fit the 3DMM to the given subject jointly, in an online fashion.

  The majority of this work has been published in [Funes Mora and Odobez, 2012]. The online fitting is yet to be published.

- **Head pose invariant 3D gaze estimation based on RGB-D cameras**. We propose a methodology which profits from depth data, and from accurate 3D head pose estimation, to rectify the eye images into a canonical viewpoint, i.e., where the appearance variations due to head pose are removed. This method effectively copes with large and unconstrained head movements, it covers well the continuous space of head poses and does not require additional training than that of a single head pose. Diverse appearance based methods in the literature can be employed in this framework.

  This work has been published in [Funes Mora and Odobez, 2012], and extended in [Funes Mora and Odobez, 2015] (under review).

- **The EYEDIAP Database**. We identified the clear need for publicly available gaze estimation datasets. The lack of such benchmarks is a serious limitation for distinguishing the advantages and disadvantages of the many proposed algorithms found in the literature. Therefore, we collected and made available to the community the EYEDIAP dataset. This database makes possible to establish a common framework for the training and evaluation of gaze estimation approaches from RGB and RGB-D cameras. In particular, we have designed this database to enable the evaluation of the robustness of algorithms

with respect to the main challenges associated to this task: i) Head pose variations; ii) Person variation; iii) Changes in ambient and sensing conditions and iv) Types of target: screen or 3D object. In total, this database is composed of 94 recording sessions.

This work has been published in [Funes Mora et al., 2014a,b].

- **Person invariant appearance based gaze estimation**. Provided the plurality of people available in the EYEDIAP database, we investigated the usage of state of the art appearance based methods to learn person invariant gaze estimation models. Two main approaches were investigated: i) to use an unsupervised sparse reconstruction framework to select subjects in a database of people whose appearance is closer to that of the test subject and; ii) to learn models combining the training data from all subjects. In addition, we discuss the inter-person eye image alignment problem which can heavily influence the performance of a person invariant gaze estimation model. An inter-person eye image alignment method, which we call *Synchronized Delaunay Implicit Parametric Alignment*, is then proposed which circumvents the detection of features, such as eye corners for this alignment task.

  The unsupervised sparse reconstruction framework has been published in [Funes Mora and Odobez, 2013]. Other elements are under review in [Funes Mora and Odobez, 2015].

- **Geometric Generative Gaze Estimation (G^{3}E)**. We proposed a new paradigm for gaze estimation from low to high resolution eye images. This approach relies on a geometric understanding of the 3D gaze action and generation of eye images. By introducing a semantic segmentation of the eye region within a generative process, the model (i) avoids the critical feature tracking of geometrical approaches which require high resolution images; (ii) decouples the person dependent eye geometry from the ambient sensing conditions, allowing adaptation to different conditions without retraining. Priors defined in the generative framework are adequate for training from few samples. In addition, the G^{3}E model is capable of gaze extrapolation, allowing for less restrictive training schemes.

  This work has been published in [Funes Mora and Odobez, 2014a,c].

- **Automatic Gaze Coding in Natural Dyadic and Group Interactions**. We demonstrated how this problem can be addressed using the proposed gaze estimation methodologies. By determining the actual 3D *LoS* and the 3D head pose of the participants, it becomes possible to automatically code gaze behavior from a simple geometric analysis. Therefore, the proposed system is based on low cost consumer RGB-D sensors. A technique to easily calibrate these sensors in a room from minimal assumptions was designed. Indeed, the challenge from these investigated scenarios is the lack of cooperation from the interactors, the need for non invasive sensing, and the unconstrained head movements.

  This work has been published in [Funes Mora et al., 2013, Oertel et al., 2014]

- ***RGBD* and *HG3D* software**. The implementation of some of the elements in this thesis have been made available to the research community. We aim at boosting gaze tracking

research and to provide a system which can work as an off-the-shelf tool for diverse fields of study and applications.

This work has been published in [Funes Mora and Odobez, 2014b]

This research was conducted in the context of the TRACOME and G3E projects, funded by the Swiss National Science Foundation.

1.4 Thesis Organization

Here we provide the organization of this thesis and briefly explain the content of each Chapter.

Chapter 2. In this literature review, we present and discuss prior methods designed to address the gaze estimation problem. We also cover prior art on head pose tracking systems due to their relevance to this thesis. We then motivate our contributions with respect to the state of the art.

Chapter 3. This chapter describes the proposed head pose estimation methodology. We cover the relevant background on 3D Morphable Models (3DMM) and registration methods and introduce the offline 3DMM fitting to RGB-D data. Assuming a personalized face model is available, the head pose tracking method is described in detail. Then an approach combining these two elements (non rigid head shape fitting and pose estimation) is described. Finally, experiments conducted on two publicly available benchmarks are presented.

Chapter 4. This chapter describes in detail the procedure used to collect the EYEDIAP database, including the annotation process which was conducted in a semi automatic manner. The structure of the data is described and possible evaluation protocols are proposed.

Chapter 5. In this chapter we will focus on the appearance based gaze estimation paradigm. We will thus describe in detail the proposed methodology intended to address head pose variations from RGB-D data. We investigate different person invariance strategies and motivate and describe in detail our proposed inter-person eye image alignment approach. Extensive experiments, conducted using the EYEDIAP database, will be presented.

Chapter 6. This chapter describes in detail the *Geometric Generative Gaze Estimation* paradigm. We cover elements such as the detailed eye geometric model, the eye image parametric segmentation function, the generative process build on top of these elements and the used inference algorithm. Experiments will be presented to validate the properties of the proposed methodology.

Chapter 7. We here describe in detail the proposed gaze coding methodology. Details about the geometric analysis and setup calibration are explained. Finally, experiments are presented on a database composed of natural dyadic interactions.

Chapter 8. Finally, we summarize the achievements of the thesis, discuss the shortcomings of the current algorithms and propose some directions for future research.

2 Related Work

In this Chapter we provide a literature review of works which are relevant to the research topic of this thesis.

As discussed in Chapter 1, head pose estimation is a requirement for most remote gaze estimation systems (cf. Figure 1.5). Humans themselves rely on knowledge of the head orientation in order to disambiguate the gaze direction, as well discussed by Langton et al. [2004]. This phenomenon has in fact been documented since the nineteenth-century and well depicted by the Wollaston Illusion, shown in Figure 2.1 [Wollaston, 1824]. Therefore, in this Chapter we will first describe prior works on computer vision based head pose estimation methods, covering different methods based on their principles and the used data modalities. This discussion will be the foundation for the method described in Chapter 3.

Following the discussion on head pose estimation methods, we will cover in detail relevant gaze estimation strategies. Finally, we will conclude the chapter by summarizing the limitations of prior works, while introducing the contributions of this thesis and how they address these limitations.

Figure 2.1: Wollaston Illusion: In both portraits the eyes regions are essentially the same. The change of the head pose nevertheless influences directly the perception of the gaze direction.

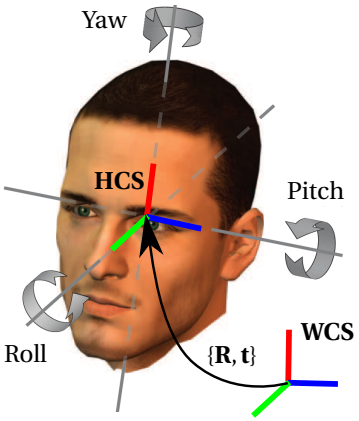


Figure 2.2: Degrees of freedom for the head pose estimation problem. The image is taken from the survey by Murphy-Chutorian and Trivedi [2008a] and modified to define the head pose as the rigid transformation (rotation and translation) relating the **HCS** to the **WCS**.

2.1 Head Pose Estimation

The problem of head pose estimation is here defined as determining the rigid transform which relates the *Head Coordinate System* (**HCS**) to the *World Coordinate System* (**WCS**), as shown in Figure 2.2. A rigid transform is composed of a rotation matrix $\mathbf{R} \in \mathbb{R}^{3\times3}$ and a translation vector $\mathbf{t} \in \mathbb{R}^3$. The rotation matrix has, in fact, only 3 degrees of freedom. These are commonly referred as the *roll*, *pitch* and *yaw* angles, and represent intrinsic rotations along the **HCS** axes, as shown in Fig. 2.2.

The following literature review is organized as follows: in Section 2.1.1 we present an overview of the many methodologies found in the literature, which follows closely the survey paper by Murphy-Chutorian and Trivedi [2008a]. Based on this overview, we will motivate the usage of *model based head pose tracking* methods, which are adequate for the main goal of gaze estimation and, therefore, are the main focus of this thesis. We will then cover different strategies used for face modelling in Section 2.1.2, followed by the methods which profit from these models to track the head under different modalities in Section 2.1.3.

The literature in head pose estimation is very abundant, thus the overview we present in Section 2.1.1 is provided mainly for completeness. However, the reader may skip this Section and focus on Section 2.1.3, which discusses the methods which are closer to the work presented in this thesis. Section 2.1.2 can be used as support material for Section 2.1.3.

2.1.1 Overview on Head Pose Estimation Methods

There are many possible criteria to characterize head pose estimation methods. Approaches differ on whether the output is discrete (coarse) or continuous (fine), on their robustness to illumination conditions variations or to facial shape variations (due to either identity or facial expresions), on whether the method requires a close view of the face or low-resolution can be

addressed, in addition to whether the approach is specific to faces or if it is applicable to other set of problems. Besides, one can distinguish among methods which leverage temporal information in a tracking framework (i.e., that require video) or those suitable for estimation from a single image. In the case of tracking, the methods further differ on how they track changes from one frame to the next, if they are susceptible to drift, and if real-time performance is possible. Finally, the degree of automation also varies a lot: some systems are fully automatic while other require manual intervention, e.g., to initialize the algorithm.

The following categorization and discussion is based in its majority on the survey by Murphy-Chutorian and Trivedi [2008a]. The authors, instead of the difficult task of classifying the many previous methods according to the characteristics discussed previously, proposed to classify them based on their fundamental principle. We include depth and RGB-D based methods, in addition to approaches based on the visual data only[1]. We also include recent methods, with respect to [Murphy-Chutorian and Trivedi, 2008a] and the development of this thesis. We will now briefly discuss each of the proposed categories:

- **Appearance based template methods** rely on the prior collection of a set of head pose annotated samples. The test data is compared to these samples through a similarity measure. Then, the output estimate is the head pose of the sample with the highest similarity to the test data. These methods provide a discrete head pose estimation, as normally the dataset is discretely annotated, or it is clustered according to head pose. Each cluster contains multiple samples with variations in terms of other quantities, e.g., identity, to improve the probability for the test sample to find a close match with the right head pose. However, the main challenge consists of designing a similarity measure which is more sensitive to head pose than to any other variable. Representative methods are: [Beymer, 1994, Sherrah et al., 2001].

- **Detector Array Methods** are based on the training of head pose specific classifiers (or detectors). These classifiers aim at learning invariances to variables which are not the head pose. This strategy also generates compact models, instead of storing and comparing a potentially large number of samples per head pose (as done in appearance based template methods). To retrieve the head pose of a test sample, all classifiers are applied to it. The assigned head pose is then the one of the classifier which outputs the maximum score. Similarly to appearance based template methods, the estimated pose is discrete and the challenge consists on developing efficient and robust classifiers, which generalize well to different identities, illumination conditions, etc. Representative methods are: [Huang et al., 1998, Rowley et al., 1998, Viola et al., 2003].

- **Nonlinear Regression Methods** aim at learning a direct mapping from the head image appearance to head pose parameters, which can be seen as an extension of detector array methods. These methods output a continuous estimate and many strategies have been

[1] By *visual data* we refer to data obtained from standard cameras which capture gray scale or color images. Alternatively it will be denoted as “RGB”, “visual domain” or “standard imaging”.

proposed. Nevertheless, generalization is a significant challenge for these methods, as it is expected for the test data to be similar to the training data and, normally, high accuracy is not achieved. Representative methods are [Murphy-Chutorian et al., 2007, Osadchy et al., 2007, Voit et al., 2007, Fanelli and Gall, 2011, Fanelli et al., 2011].

- **Manifold Embedding Methods** apply dimensionality reduction techniques to the high dimensional head image. Then, similar to the previous methods, a regression algorithm is used to learn the mapping from the low dimensional image representation to the head pose parameters. The main assumption is that, in principle, the set of all possible head images lie in a lower dimensional manifold of which the head pose is responsible for most variations. Dimensionality reduction methods, such as principle component analysis (PCA), linear discriminant analysis (LDA) and their kernel-based versions are common. The main limitation of these methods is that there is no guarantee for the low dimensional representation variations to be indeed mainly due to head pose; even though supervised alternatives have been proposed to reinforce the focus on head pose parameters. Representative methods are [Srinivasan and Boyer, 2002, Raytchev et al., 2004, Wu and Trivedi, 2008, Yan et al., 2008, BenAbdelkader, 2010, Wang and Song, 2014].

- **Flexible Models** are based on the prior definition (or learning) of face models which are fit to the test data. This fitting may rely on the prior localization of facial features, whose spatial configuration is compared to a predefined set of head pose configuration, or may rely on the fitting of statistical parametric face models to data. To this end, normally a fitting by synthesis is conducted, i.e., the cost function evaluates the discrepancies between the data and the model instance generated by the current estimate of the model parameters. These models are therefore generative and can be learned from collections of data, as discussed in Section 2.1.2. Generalization to unseen individuals is an important challenge for this strategy, in addition to low resolution imaging and missing facial features handling, due to occlusions or out of plane rotations. Representative methods are: [Lanitis et al., 1995, Cootes et al., 2000, Krüger et al., 1997, Xiao et al., 2004b, Smolyanskiy et al., 2014].

- **Geometric Methods** are the approaches which are closer to the human perception system of head pose [Wilson et al., 2000]. These methods rely on the prior localization of facial features. The 3D head orientation can then be determined from their positions, either from relative distances or by minimizing the distance between the landmarks estimates and the projection of a 3D model of facial landmarks positions, defined with respect to the **HCS**. This model can be fixed, based on anthropomorphic averages, or it can be tuned to the given subject. The main challenge for these methods consists of accurately determining the facial landmarks and to address facial expressions. Representative methods are: [Gee and Cipolla, 1994, Horprasert et al., 1996, Wang and Sung, 2001, 2007]. Notice that, even though these methods have normally required high resolution data, recent advances have shown promising accuracy on landmarks localization under lower resolution conditions and far from frontal head orientations: [Dantone et al., 2012, Zhu and Ramanan, 2012,

Xiong and De la Torre Frade, 2013, Cao et al., 2013, Kazemi and Sullivan, 2014].

- **Tracking Methods** assume the data is provided as a time sequence (video). The goal becomes to track the change in pose between succesive frames. These systems normally report higher accuracy than previous categories [Murphy-Chutorian and Trivedi, 2008a]. Many methods have been proposed, which can be based on features [Gee and Cipolla, 1996, Yao et al., 2001, Ström, 2002, Yang and Zhang, 2002, Zhao et al., 2007, Jang and Kanade, 2008, Wang et al., 2012], predefined models [Pappu and Beardsley, 1998, Schödl et al., 1998, Wu and Toyama, 2000, Cascia et al., 2000, Xiao et al., 2002, Lefèvre and Odobez, 2009], dense optical flow [Morency et al., 2002], particle filters [Oka et al., 2005, Murphy-Chutorian and Trivedi, 2008b] and surface registration [Weise et al., 2011]. We will discuss some of these methods in more detail in the following Sections.

- **Hybrid Methods** combine some of the previous methodologies to profit from their advantages and overcome their shortcomings. There are many proposed methods that can be set in this category: [Horprasert et al., 1997, Jebara and Pentland, 1997, Sherrah and Gong, 2001, Morency et al., 2003b, S.Huang and Trivedi, 2004, Ba and Odobez, 2004, Murphy-Chutorian and Trivedi, 2008b, Wu and Trivedi, 2008, Sung et al., 2008, Morency et al., 2010, Baltrusaitis et al., 2012]. We will discuss some of these methods in more detail in the following Sections.

The previous categorization is useful as it allows to identify adequate head pose estimation strategies for gaze estimation. In particular, it is required for a head pose tracker to provide continuous and accurate estimates. Methods based on an explicit model of the face are convenient, as the frame by frame eye localization would be solved by the head pose tracker, as long as the eye position can be defined as a fixed translation within the **HCS**. Therefore, according to these requirements, we discard appearance templates and detector arrays based methods, as they output discrete estimates of the head pose. Non linear regression and manifold embedding methods are also not adequate for gaze estimation, as they normally report low accuracy. Furthermore, none of these methods provide explicit information on the eyes location.

Flexible models and geometric methods meet better these requirements, in particular when used in a tracking framework. As mentioned previously, we will focus on model based methods, which can exploit characteristics from both categories and that we now describe in more detail. First, the different face modelling strategies will be described, followed by the tracking algorithms, which build on top of these models.

2.1.2 Face Modelling

Face representation is a crucial element for model based head pose tracking methods. In this section we will first discuss the main face modelling strategies found in the literature, from rigid models to flexible representations which can address different people and facial

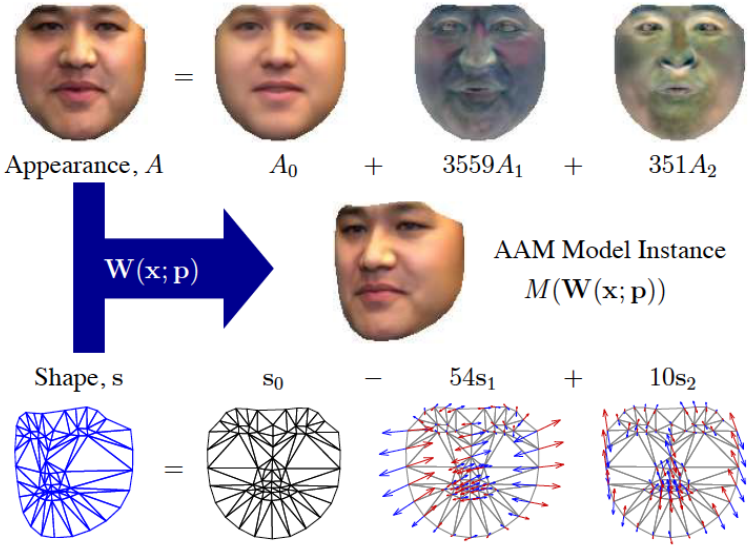


Figure 2.3: Active Appearance Models instance generation. Notice that the mean shape is linearly combined with deformation vectors. Similarly, the texture instance is generated in a normalized frame and then warped to the shape instance [Matthews and Baker, 2004].

expressions. Then, in in Section 2.1.3 we will describe how these models have been used in a 3D tracking framework.

2D Face Models

Since the work of Cootes et al. [1995], linear subspace techniques have been a major approach to represent faces and their deformations. *Appearance shape models* (ASM) were introduced to represent the 2D shape of a class of objects. Provided shape is first defined as a set of 2D points, normally associated to a predefined topology, ASMs represent the class of objects as the sum between the average shape and a linear combination of a 2D basis of shape deformations. Notice that, in the case of faces, the deformation basis can model variations in shape across people, facial expressions or even small head pose variations. Typically, principal component analysis (PCA) is applied to a corpus of annotated training images, leading to the extraction of eigenshapes as the deformation basis.

Active Appearance Models (AAM) were later introduced as well by Cootes et al. [1998]. The authors proposed to define also a linear subspace representation for the face texture, in addition to the shape. Provided the annotated data texture is first piece-wise warped into the mean shape, PCA is applied to the corpus of rectified face images, generating a mean texture and eigenfaces [Turk and Pentland, 1991] as a texture variation basis. The result is a powerful generative statiscal model able to represent both the shape and appearance of people, as depicted in Fig. 2.3.

ASM and AAM have been shown to be powerful approximation tools to represent 2D faces and,

as a main advantage of their linearity, relatively simple algorithms with low complexity have been demonstrated to fit efficiently and precisely any new near-frontal face under reasonable illumination conditions and resolution from a coarse initialization [Matthews and Baker, 2004]. In addition, semi-automatic strategies have been proposed to facilitate the model annotation/learning process [Ramnath et al., 2008].

Still, linear shape models have encountered difficulties in approximating shape deformation due to pose changes, especially when they reach the point where facial features get self-occluded (beyond 45 degrees). This is a known limitation for these models. Nevertheless, extensions have been proposed to alleviate this problem. Cootes et al. [2000] proposed to use multiple pose-specific AAM, covering the range of head poses. Alternatively, Gross et al. [2006] proposed an occlusion handling framework when fitting an AAM to data. This strategy implicitly address large head poses and their associated self occlusions.

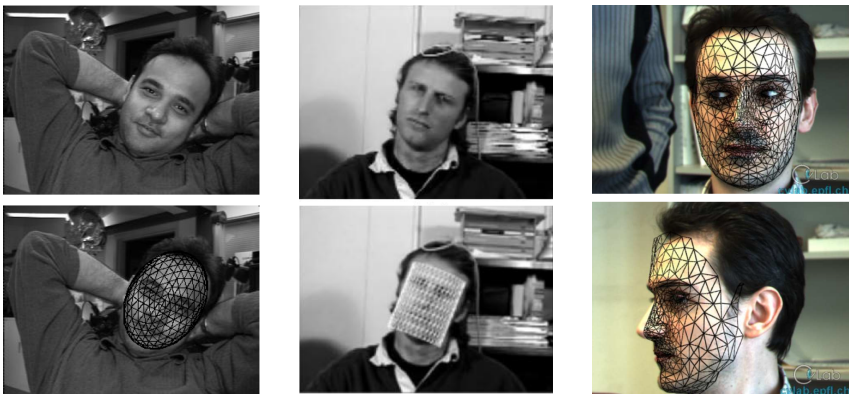
In this thesis the end goal is to retrieve the 3D head pose, and AMMs can be used to this purpose. Notice that AAM implicitly extract the 2D location of facial features. Therefore, the 3D head pose could be computed from their position using geometric methods, which nevertheless would require a priori a model of the 3D features location. Alternatively, Xiao et al. [2004a], relying on structure from motion techniques, proposed a methodology to create this model from a sequence of AMM fitting. Furthermore, the authors also proposed to introduce the 3D model as a constrain to the 2D AAM fitting, leading to a 2D fit which is feasible in terms of 3D head pose. Although this algorithm retrieves the 3D head pose measurements, a 2D tracking sequence is first required.

3D Face Models

3D face representations have been proposed to address larger head poses, which are challenging for 2D face modelling strategies. These strategies differ on whether they model the face as a rigid or non-rigid object, on how accurately they approximate the face shape, and whether they are generic or person specific. In addition, there are differences in terms of the way appearance is represented, which can be crucial to cope with significantly far from frontal head poses, to handle drift or to gain robustness to illumination variations.

Rigid representations. Basu et al. [1996] initially proposed to use a 3D ellipsoid to approximate the face shape, as seen in Fig. 2.4a. This was demonstrated to be a good approximation, in contrast to planar models. Bregler and Malik [1998] used later the same modelling strategy to represent each of multiple body segments. Other authors, such as Malciu and Prêteux [2000] and Morency et al. [2008] have also made use of this strategy. Notice that the aforementioned methods do not have an explicit appearance representation.

Alternatively, Cascia et al. [2000] proposed to use a cylinder to approximate the face shape. Although this was a looser fit in comparison to the ellipsoid, the cylinder face model was augmented with the texture captured from a reference image. In addition, it was also pro-



(a) (b) (c)

Figure 2.4: Rigid face models. (a) Ellipsoid [Basu et al., 1996]. (b) Cylinder [Cascia et al., 2000]. (c) Detailed mesh [Vacchetti et al., 2004].

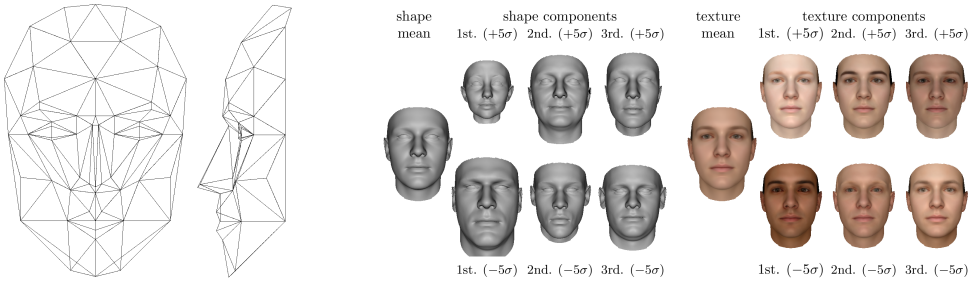
posed to linearly combine the reference texture with a person independent basis of texture illumination variations. This representation can help to cope with the texture changes which naturally occur due to head pose variations. Xiao et al. [2003] relied on the same face modelling, nevertheless, within a tracking framework, they proposed to use a dynamic texture, as this was demonstrated to be important for the appearance representation under illumination variations and, more importantly, under larger head poses, whose appearance viewpoint might differ significantly to a single reference texture.

More accurate 3D face rigid mesh models have been also used, as seen in Fig. 2.4c. Nevertheless, these models are normally built offline, in particular, the ones accurately modelling the shape of the person to be tracked [Schödl et al., 1998, Vacchetti et al., 2004, Weise et al., 2011].

The modelling of the 3D face appearance can be enriched by keeping a set of *keyframes*. This corresponds to images of the object of interest captured under different poses. This allows to have a global and absolute representation of the object of interest which, within a tracking methodology, may prevent drift. In the work by Vacchetti et al. [2004], the set of keyframes and their poses were collected offline and the 3D mesh was needed to determine the relation between 2D feature correspondences, based on the object's pose. Alternatively, Morency et al. [2003a] proposed a strategy to collect keyframes online, while addressing the challenge of refining the estimated pose of these keyframes, according to new observations.

A limitation of these models, is that they are restricted to a rigid face representation, both in terms of shape and appearance. In addition, the obtained **HCS** is not necessarily semantically consistent between tracking sessions, meaning the position of the eye can change. This can nevertheless be alleviated by maintaining the same appearance representations accross sessions.

Non-rigid 3D face parametric models. *3D Morphable Models (3DMM)* have been proposed



(a) (b)

Figure 2.5: Non-rigid parametric 3D face models. (a) Candide face model (mean shape) [Rydfalk, 1987]. (b) Basel Face Model. Showing the shape and texture variations separately [Paysan et al., 2009].

as a direct 3D extension of ASMs and AAM. These models are able to represent the face shape variations as a linear subspace, with a relatively small set of coefficients. Similarly to ASMs, shape in a 3DMM is modelled as a set of 3D points, for which the mean shape is available, together with the deformation basis. Normally, an associated topology establish the connectivity among the 3D points, thus defining shape as a 3D mesh. In the same way to AAM, appearance is modelled as a mean texture and a texture variations basis, within a shape-normalized frame. As an early work, Vetter and Blanz [Vetter and Blanz, 1998] built such 3D morphable model (3DMM) to extract 3D face models from single images.

These models can be defined manually (shape in particular). A classic example is the Candide face model [Rydfalk, 1987], for which the mean shape (see Fig. 2.5a) and deformation basis were defined by an artist. Alternatively, 3DMMs can be learned automatically from collections of 3D head scans or 3D facial landmark positions, extracted using multi-view systems [Vetter and Blanz, 1998, Göktürk et al., 2001, Zhang et al., 2008, Paysan et al., 2009]. ASM and AAM can be applied to automatically establish the correspondences between the set of scans or, alternatively, non-rigid 3D registration methods can be used to define denser correspondences [Amberg et al., 2007]. Once the scans are well registered, principal component analysis is normally used to extract both the shape and texture basis. A prominent example, available for research purposes, is the Basel Face Model, developed by Paysan et al. [2009], and depicted in Fig. 2.5b. This model was learned from high quality 3D scans of 200 subjects, 100 male and 100 female. The result is a powerful model capable of spanning a large set of identities.

Identity and expressions modelling. One general issue with 3DMMs (also valid in ASMs and AAMs) learned from training data is that two origins for shape variations are mixed in the same model: different people and different expressions. This leads to larger models, which require to estimate a large number of parameters when addressing a tracking task. Nevertheles, in principle only the expression components would vary during tracking.

A solution to this issue is to represent the model's shape with separate and linearly combined

manifolds for identity and expressions. These subspaces can again be either learned separately [Liao and Medioni, 2008, Amberg et al., 2008, Cosker et al., 2011] or defined manually, e.g. from Ekman's Facial Action Coding System (FACS) [Ekman and Friesen, 1978], by relying on an artist [Smolyanskiy et al., 2014]. The advantage of a FACS basis is its semantic interpretation, which is not given by PCA (directly), and for which there is a direct link to expression analysis and animation. Nevertheless, to improve the accuracy of the model, the deformation basis can be made specific to a given user [Gross et al., 2005], e.g., in a semi-supervised manner or in a user cooperative manner [Li et al., 2010, Weise et al., 2011]. Notice that strategies, such as bilinear factorization methods, have also been proposed to model identity and expressions, as these may capture dependencies between the identity and deformation basis [Abboud and Davoine, 2005], which are ignored when using independent representations.

2.1.3 3D Head Pose Tracking

In this section we describe how prior works have made use of face/head models (cf. Section 2.1.2), for the 3D head pose tracking task. The main goal of head pose tracking is to retrieve the 3D head pose parameters frame by frame within a sequence. Nevertheless, depending on the model, the task may further require the estimation of deformation and appearance parameters, or alternatively, to redefine the model itself online to improve the tracking.

Tracking strategies are dependent on the modality. For this reason, we will first cover the visual domain alone, then briefly discuss stereo and RGB-D modalities.

Visual domain based tracking

Many 3D head pose tracking methods rely on the frame by frame fitting of 2D ASM and AAM, which we will describe as follows. The fitting of ASMs often rely on adjustments of the model points such that they match image edges, while being constrained by the plausible ASM deformations and its overall rigid transform. This problem is equivalent to optimizing an energy function which was applied to faces by Lanitis et al. [1997].

AAM based 2D face tracking. The per-frame fitting AAM problem consists on minimizing the discrepancy between the image data and the face image generated by the AAM. Image based gradient descent extensions of the Lucas-Kanade algorithm were developed to refine the model parameters estimation until convergence [Cootes et al., 1998, 2000, Baker and Matthews, 2004]. Later, Matthews and Baker [2004] proposed the inverse compositional algorithm which, by defining the change of parameters as a composed transformed (rather than additive) and inverting the warp direction, is able to precompute many of the elements needed during gradient descent. The resulting algorithm is much faster, allowing for real-time AAM fitting implementations.

The advantage of AAM approaches is that very precise face rendering and registration can be obtained, but, as AAM are sensitive to illumination conditions, the aforementioned opti-

mization techniques require a good initialization to ensure a correct convergence. In addition, these models are usually best employed in conditions similar to the training data. In particular, AAM are much more accurate when created as person-specific models, as evaluated by Gross et al. [2005].

However, these models have difficulties to address large out of plane head pose variations. Thus, it is common to resort to multiple pose-specific AAM, known as *view-based* AAM. As proposed by Cootes et al. [2000], during the tracking, the selection of which pose-specific AAM to use can be done by monitoring the current head pose. Alternatively, out of plane rotations can be addressed by tackling self occlussions, as done by Gross et al. [2006] which reformulated the registration error to discard outliers using a robust estimator.

AAM based 3D head pose tracking. Notice that a 2D AAM implicitly provides a set of 2D features. Therefore, the 3D head pose can be inferred by minimizing the projection error of a 3D face model. This strategy would follow the "Geometric Methods" paradigm described in Section 2.1.1. However, this requires for the 3D face model to be defined a priori. Alternatively, as proposed by Xiao et al. [2004a] with the 2D+3D AMM formulation, a 3D morphable model can be used for both constraining the AAM fitting to lead to 3D plausible solutions and to estimate the 3D head pose accordingly.

Zhou et al. [2010] later proposed to introduce additional terms to the 2D+3D AAM fitting and view based AAM framework: temporal matching and facial segmentation constraints. The former term is used to foster model parameters which maintain appearance consistency between local patches at consecutive frames. The latter term is used to disambiguate between the face region and background/outliers by using face segmentation based on adaptive color models. The resulting tracking is more robust to fast movements and clutter.

3DMM based 3D head pose tracking. Note that 3DMM can be directly fitted to 2D image data, as done by Vetter and Blanz [1998], Blanz and Vetter [2003], with a similar analysis by synthesis scheme to AAM, i.e., minimizing the error between the data and the model generated face image. Nevertheless, the combination of rigid motion and perspective projection model lead to a non-linear relationship between the model parameters (pose, deformations) and the image information, which leads to complex and potentially unstable fitting. Therefore, instead of fitting the 3DMM directly, ASMs and AAMs can be used to first reliably track 2D features. The 3DMM can then retrieve the face shape deformations and the 3D head pose from the 2D features. Moreover, the 3DMM may be used to constrain the 2D features tracking such that their final localization is coherent with plausible head poses, as done by Xiao et al. [2004a], Vogler et al. [2007]. Alternatively, particle filter based approaches have been used to account for multiple hyphothesis, as used by Dornaika and Davoine [2008] when inferring 3DMM deformations.

Constrained local models (CLM). These models, proposed by D.Cristinacce and T.F.Cootes [2007], are an alternative to AAM (2D). Similarly to AAM, they model shape and appearance as linear subspaces. However, the appearance is represented by patches around the shape

points, instead of by the whole face. Normally these patches correspond to semantic elements, like the eyes, mouth corners, etc. The fitting strategy is radically different to AAM, as it is based on the optimization of patches filters responses constrained by the shape model, thus it does not need to jointly explain the entire face image appearance. Empirically, CLM were found to be more accurate than standard AAM. However CLM may suffer from self occlusions under large head poses, in which many of the face elements are no longer visible.

Rigid models based 3D head pose tracking. Methods which make use of 3D rigid models are based on different strategies than AAM and CLM. In general, these methods exploit differences in appearance between frames, which are assumed to be only due to pose changes. This may further rely on creating appearance models associated to poses, collected offline or online.

Basu et al. [1996] and Malciu and Prêteux [2000] formulate the tracking problem as inferring the sequence of rigid transformations of the model which best explain measurements on optical-flow. Note that this does not require explicit appearance modelling. Nevertheless, this approach is succeptible to drift, or error accumulation.

To avoid drift, Cascia et al. [2000] proposed to extract the face texture from the first frame and map it into a cylindrical model. The problem of tracking was then formulated as finding the 3D pose parameters of the cylinder which would minimize the appearance difference between the image data, and the textured cylinder. As mentioned in Section 2.1.2, the appearance model can be enhanced to take into account illumination variations. Nevertheless, a single texture may lead to appearance mismatches at larger poses. Therefore, Xiao et al. [2003] proposed to dynamically adapt the texture. In practice, the first frame's texture and a few others at diverse head poses are maintained to rectify the accumulated errors, preventing drift.

To build an appearance model suitable for tracking under very large pose variations, Vacchetti et al. [2003, 2004] proposed to rely on keyframes (see Section 2.1.2) collected offline and preprocessed to accurately know their 3D poses. The tracking problem is formulated as minimizing the distance between a set of *interest points* and their established correspondences -back and forward- projected through the rigidly transformed 3D model of the object (defined a priori). The optimization comprises both the offline data (keyframes) and past frames in order to decrease jitter and to prevent drift. Alternatives have been proposed to make the difficult problem of interest points matching faster and more robust. Lepetit et al. [2004] proposed to treat the matching as a classification problem, in which a class corresponds to a single interest point, but under many appearance variations, e.g., due to viewpoint. This framework was futher improved and tailored for faces by Wang et al. [2012].

Morency et al. [2003a] proposed an approach for the online collection of keyframes. The method relies on an algorithm to estimate the pose change between two frames. Initially it is used to do a differential sequential tracking. However, whenever certain poses are observed, these frames and their pose are added to a collection of keyframes. Once keyframes are available, the tracking comprises both the previous frame and the keyframes set. Based on a gaussian linear filter, the approach further refines the pose of both the input data and the

keyframes, improving the overall tracking. This work was initially applied to stereo data and then later refined [Morency et al., 2008] to handle monocular video sequences, and further combining with static head pose estimation to improve the tracking accuracy. The authors called this framework generalized adaptive view-based appearance model (GAVAM).

RGB-D and stereo based tracking

3D representations of the head data can be obtained from depth imaging (or stereo). While this representation can be explicit, in the form of a 3D mesh, it has also been proposed to use regression methods to infer a mapping directly from the depth image to head pose parameters [Fanelli et al., 2011, Fanelli and Gall, 2011]. Even though good generalization has been achieved using these methods, semantic information is lost and low accuracy is normally achieved.

With an explicit representation of the 3D data, obtained from stereo data or from depth sensors, the well known iterative closest points (ICP) algorithm can be used. ICP was initially proposed by Chen and Medioni [1991] and Besl and McKay [1992]. It is used for the registration of 3D surfaces, i.e., finding the pose that best aligns one surface to another. Rusinkiewicz and Levoy [2001]'s survey cover the many variants of the algorithm.

ICP can be used as a differential tracker, i.e., to estimate the change of pose by registering successive frames, as done by Morency and Darrell [2002]. In this work, the authors further proposed to include normal flow constraints, as these are useful to relate the changes in image appearance to the object velocity (i.e., change on pose parameters). This can help alleviate poor ICP estimates under low quality data. The authors thus proposed to estimate the change of pose parameters as the least squares solution of a linear system of equations, embedding both the ICP registration and the normal flow constraints. Empirically, the authors found that the combination indeed leads to a more robust and accurate tracking.

Alternatively, a template mesh of the face can be created offline, and registered to each frame [Weise et al., 2011]. Often, the previous pose estimation is used as initialization parameters for the next frame. In the work of Weise et al. [2011], the template is built offline using a non rigid registration method which requires user cooperation, in order to collect appropriate data. Very accurate head pose estimates can be obtained with this method, even when applied to a consumer depth sensors (Microsoft Kinect), but it depends on the accuracy of the template, which can limit the application of this strategy to unseen users.

A limitation of ICP is its decrease in accuracy in the presence of facial deformations. Nevertheless, as done by Weise et al. [2011], this can be alleviated by using only the upper part of the face for the tracking, where non rigid deformations are minimal. Alternatively, non-rigid parameters could be computed by extending the template into a 3DMM and using non-rigid ICP variants, such as the method proposed by Amberg et al. [2008]. Weise et al. [2011] also fit an expressions deformation model during tracking, constrained by a deformations prior model. Optical flow is used as further observations to obtain the expression coefficients. In

practice, the facial deformations are obtained after the head pose has been estimated using the rigid ICP.

Baltrusaitis et al. [2012] extended the CLM framework by including depth patches observations, in addition to the standard visual domain patches, leading to what they called the CLM-Z model. This approach provides better support, as some regions can be ambiguous to one modality, but better discriminated in the other. Furthermore, depth patches can be used in conditions of poor illumination. The CLM-Z approach was demonstrated to be better than the CLM or GAVAM approaches. Combined with the GAVAM framework to obtain a rigid+non-rigid face tracking approach, it demonstrated further improvements.

Bouaziz et al. [2013] later proposed to extend the framework of Weise et al. [2011] to include the online fitting of an identity 3DMM, together with the online refinement of the expressions model, bootstrapped by a set of manually defined deformations, or blendshapes. This reduced the amount of needed user cooperation.

Very recently Smolyanskiy et al. [2014] extended the AAM framework proposed by Zhou et al. [2010] to include depth constraints as a new term in the cost function, which basically corresponds to a point-to-point ICP. It is intended to improve the 3D face localization, in particular, along the depth axis. The authors demonstrated that 2D AAMs are innacurate along this dimension, especially because the true object's size is undetermined with monocular cameras. Their model included a 3DMM with separate shape (identity) and animations (expressions) basis as well. The shape parameters are fit in an initialization phase.

2.2 Gaze estimation

As described in detail in Section 1.2, the gaze estimation problem consists of retrieving either the 3D line of sight (LoS) or the 2D/3D point of regard (PoR). Normally, a gaze estimation system includes an eye localization step, where its goal depends on the gaze estimation methodology and data modality. However, as mentioned previously, in this thesis we will focus on model based head pose tracking. In this context, the eye position can be defined as a fixed point within the **HCS**, from which the frame by frame eye localization is estimated from the head pose parameters. Therefore, the head pose tracking implicitly solves the eye localization problem[2]. Furthermore, depending on the gaze estimation method, the exact position of the eyeball within the **HCS** might not be required, or its position can be refined by the gaze estimation algorithm itself during a calibration phase.

The automatic estimation of gaze has been investigated for over three decades, as well described in the comprehensive survey by Hansen and Ji [2010]. In this Section we will focus on the two main methodologies which can be identified: *geometric based* methods and *appearance based* methods.

[2] For an overview on alternative eye localization methods, please refer to [Hansen and Ji, 2010].

(a) (b)

Figure 2.6: IR illumination and sensing for gaze tracking. (a) Dark/bright pupil effect under IR illimination [Hansen and Ji, 2010]. (b) PCCR features: pupil-center and corneal-reflection(s) [Guestrin and Eizenman, 2006].

2.2.1 Geometric based gaze estimation methods

These methods rely on the detection of local features which are mapped to gaze parameters. Therefore, in much of the literature these methods are called *feature based methods*. Most methods require a calibration session to collect gaze annotated samples. These are used to determine user specific parameters, describing the eyeball geometry, or to derive a mapping from the features location into the point of regard, specially for screen based scenarios.

Infrared based methods

The most accurate techniques in this category rely on infrared (IR) illumination and sensing. The main advantage of working in the IR spectrum is the capture of eye images which exhibit high pupil/iris contrast, under which it is easy to estimate the pupil center. Moreover, the dark/bright pupil effect, as seen in Fig. 2.6a, may be exploited to detect the pupil. It is also common to add IR light sources to the setup, as these generate specular reflections along the cornea surface. These reflections, commonly referred to as *glints*, are also relatively easy to detect in IR images (see Fig. 2.6b). This methodology is known as *PCCR* (pupil-center corneal-reflection) [Morimoto and Mimica, 2005] and it is the most popular sensing technique used in commercial gaze tracking systems ([SMI, 2007, EyeGaze, 2005]). Notice, illumination under this spectrum is invisible to the human eye and it is kept at power margins safe for deployment on commercial products. There are different strategies to infer gaze from PCCR data, but they can mainly be categorized as *interpolation* methods and *eye model based* methods.

Interpolation methods apply a direct mapping from these features to the PoR position. Normally, the PoR corresponds to a 2D point within a flat screen. Many strategies were proposed, but the main challenges have consisted on the learning of a mapping which could be invariant to the non linear movement of the features along the eyeball/cornea surface, and to head movements. Therefore, to this end, many strategies were proposed, including linear interpolation [Merchant et al., 1974], polynomial regression [Morimoto et al., 2000], dynamic mappings to account for head movements [White K.P. et al., 1993] and non linear mappings, including Neural Networks, Support Vector Regression [Zhu et al., 2006] or Gaussian Processes (GP),

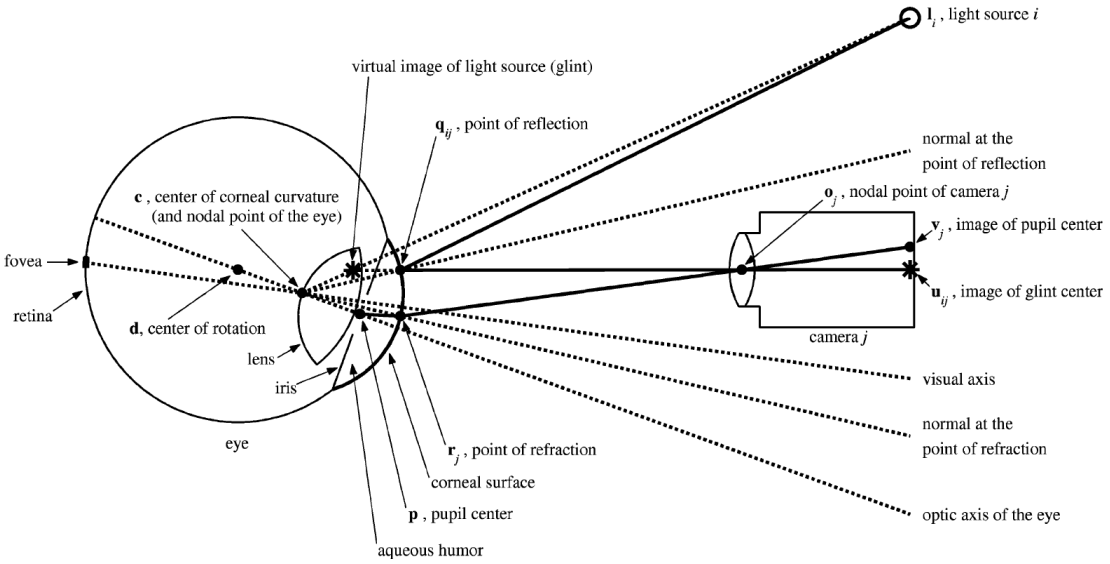


Figure 2.7: Geometric eye model based gaze tracking. Notice the explicit modelling of the system geometry, including the camera, light sources and the eyeball. Image taken from Guestrin and Eizenman [2006]

whose covariance estimates may be used to detect discrepancies between the input data and the calibration data, e.g., due to head movements [Hansen et al., 2002].

Eye model based methods, on the other hand, make an explicit geometric modelling of the human eyeball and the setup configuration, as shown in Fig. 2.7. The theory behind this methodology is well known, and described in detail in the paper by Guestrin and Eizenman [2006]. The authors derive the geometrical relations between the PCCR measurements, the eye model and the light sources, but, as they demonstrate, the setup configuration is crucial. If a single camera and single light source is used, gaze can be estimated for a single head pose only. By adding multiple light sources to a single camera setup, it is possible to compute gaze under head pose variations, as shown by Shih et al. [2000], but a calibration session is required by fixating at multiple points. If multiple cameras and multiple light sources are used, it then becomes possible to infer gaze from a one point calibration procedure and under head pose variations.

Cross-ratio methods, which were initially proposed by Yoo and Chung [2005], make use of the careful placement of light sources at the 4 screen corners. By measuring the produced glints positions, including the glint of a 5th light source along the camera axis (which also produces the bright pupil effect), these methods profit from the projective invariaces defined between the 4 LEDs in the screen plane, the camera plane and a tangential plane to the cornea. By measuring the pupil center displacements, they are able to relate its position to screen coordinates. Notice, however, this methodology is only suitable for user computer applications.

Homography normalization, proposed by [Hansen et al., 2010], is an approach suitable for uncalibrated setups. The method is based on a similar approximation to cross-ratio methods, which assumes that the glints reflection are coplanar within the corneal surface. It then models the set of light sources (at the screen), the corneal reflections (glints) and their projection at the camera as a set of planes, related by a sucession of homographies. The method is fairly robust to head pose changes, it is able to model the offset between the visual and optical axes (as opposed to cross-ratio methods) and require only a few calibration points (9 to 16). Its performance was empirically shown to be better than cross-ratio methods.

Overall, the main disadvantage of IR based gaze tracking systems is the need for specialized setups and high resolution eye images, in order to detect the PCCR features. This leads to costly hardware and/or constrained range of operation, and would be really difficult, if not impossible, to use for our human-human interaction analysis scenarios.

Natural light based methods

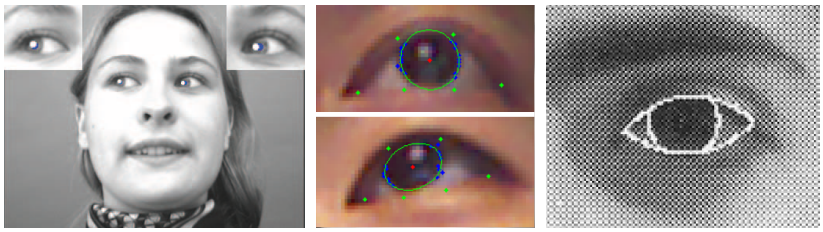
Under natural light conditions, many proposals also leverage local eye features to build geometric models of the eyes. Features such as the iris center, retrieved through voting techniques based on gradient features [Timm and Barth, 2011] or isophote features [Valenti and Gevers, 2012], an ellipse fitted to the pupil/iris, e.g., using the Starburst algorithm [Li et al., 2005] or active contours relying on bayesian tracking techniques (e.g., particle filters) [Hansen and Pece, 2005], or complex shapes incorporating the eyelids contours [Yuille et al., 1992, Wang and Sung, 2002] or even the full eye region [Moriyama and Cohn, 2004] could be used. Examples are shown in Fig. 2.8.

Once the eye features are located, the goal consists on estimating the *LoS*. Valenti et al. [2012] proposed to use a direct mapping from the 2D iris center into a gaze direction within what they called the subject's field of view. This field of view is constantly redefined according to the estimated head pose. Isophote features were used to locate the iris, but this search was done over an eye image which is pose-normalized, i.e., it is rectified into a frontal looking face image based on texture warping through a cylindrical head model[3].

Other authors rely on eye geometric models to estimate the gaze direction. Ishikawa et al. [2004] proposed to find the eyeball geometric parameters through a specific, but not flexible, calibration protocol. This protocol was designed to relate the calibration iris center measurements (obtained through ellipse fitting) and the fixated points with the person specific eyeball geometry. Active appearance models were used to track the head pose, and thus to obtain the eyeball center under head pose variations. Once the eye model parameters were found, gaze could be obtained from the ellipse fitting and head pose tracking at test time.

Yamazoe et al. [2008] proposed a methodology to avoid explicit cooperation from the par-

[3] In Chapter 5 we propose a warping methodology based on depth measurements or more accurate 3D face models.



(a) (b) (c)

Figure 2.8: Local eye features. (a) Iris center [Valenti and Gevers, 2012]. (b) Ellipse fitting [Li et al., 2005, Xiong and la Torre, 2014]. (c) Deformable parametric eye model [Yuille et al., 1992].

ticipant. Their strategy, at calibration time, relied on obtaining the eyeball geometric parameters which would maximize the pixel-wise sclera and iris classes agreement between pre-segmented eye images and their segmentation based on the eye geometric model. Facial features detection were used to create a person specific facial landmarks model and for the head pose tracking. Ellipse fitting was also used but obtained from the segmentation of the iris region based on thresholding.

RGB-D based methods

Recent methods were proposed to apply the same geometric based gaze estimation methodology to RGB-D data[4]. Jianfeng and Shigang [2014] used a Microsoft Kinect™of first generation to this end. The authors relied on the iris center localization algorithm by Timm and Barth [2011], whereas the head pose tracking was done using the method available within the Microsoft Kinect's SDK. The eyeball center is refined from a calibration session, whereas the rest of eyeball parameters are not learned. Xiong et al. [2014] used the same sensor, but relied on a ellipse fitting algorithm based on the Starburst method [Li et al., 2005], and facial landmarks for 3D head pose tracking, as well as to build a person specific facial landmarks position model. Their calibration method infers additional eyeball parameters. However, in both cases, the Kinect had to be configured for the highest RGB resolution of 1280×960 and the participant had to be close to the sensor, in order to track the local eye features. Notice that, in both systems, depth data was only used for the purpose of head pose tracking.

Overall, an important limitation of geometric based methods is the need to detect the local features, as it requires high resolution and high contrast images. This is not a limitation of appearance based methods, which are discussed in the following Section.

2.2.2 Appearance based gaze estimation methods

By modeling a direct mapping from the entire eye image to gaze parameters, these approaches avoid the local features tracking task. This methodology has therefore potential for low-

[4]The initial proposal to rely on RGB-D data for the gaze estimation problem is one of the contributions of this thesis [Funes Mora and Odobez, 2012].

resolution gaze sensing.

As a pioneering work, Baluja and Pomerleau [1994] relied on an artificial neural network to map directly from the eye image pixels, defined as the network's inputs, to screen coordinates defined as discrete outputs. However, in their experiments they required > 2000 training samples to obtain acceptable gaze estimation accuracy. Furthermore, their system required a fixed head pose, as otherwise, they mentioned their system would require much more training data. Nevertheless, this was a common requirement then, even for IR based systems.

Standard gaze estimation task

Following the work of Baluja and Pomerleau [1994], other authors proposed alternative methods to be applied under similar conditions, i.e., minimal head pose variations and allowing in-session calibrations. Tan et al. [2002] proposed to use linear interpolation to reconstruct a test sample from a local appearance manifold within the training data. To select the samples to interpolate from, instead of using techniques such as k-Nearest Neighbors, the authors proposed to exploit the topology information, encoded in the 2 dimensional space of gaze parameters. A delaunay triangulation was therefore used to represent this topological information and to constrain the samples selection. Their method effectively reduced the training samples to 252 while achieving good accuracy.

Lu et al. [2011a], as an alternative to the method of Tan et al. [2002], proposed to apply the linear interpolation of the test sample using all samples in the training set, with the additional constraint of enforcing sparsity on the reconstruction weights. Experimental results demonstrated that sparsity had the implicit function of selecting samples within a local appearance manifold. The authors report high accuracy, even when using small training sets. However, the experiments were conducted under carefully controlled conditions, such as requiring a fixed head pose (using a chin-rest), same illumination settings and well aligned eye images. A methodology based on sparsity was also proposed to compensate for minor eye image translations in their experiments.

Soft calibration methods

To further reduce the required amount of calibration samples, other authors have proposed to use weakly annotated data. Along this direction, Williams et al. [2006] proposed a semi-supervised sparse gaussian process regression (S^3GP) method. The main idea is to profit from the samples which are observed during a gaze shift, occurring between successive calibration points. Their method achieved good accuracy with only 16 calibration points.

Alternatively, Sugano et al. [2008] proposed to exploit user-computer interaction traces as training data, instead of explicit calibration sessions. In addition, to address head pose variations, the authors proposed to create separate gaze appearance manifolds clustered according to head pose. At each manifold, softly selected based on head pose, the same

technique proposed by Tan et al. [2002] was used to infer gaze. However, the amount of required training samples and the gaze estimation error heavily increased, due to head pose variations.

Head mounted setups

Head mounted setups are of interest as they allow for unconstrained head movements while capturing high resolution eye images under a single viewpoint, therefore, facilitating the estimation of gaze. Within the appearance based paradigm, Noris et al. [2010] designed a head mounted system intended to be used with small children. The authors used Support Vector Regression (SVR) [Smola and Schölkopf, 2004] to map from eye images into the fixation point of an egocentric video. The images are first processed using a weighted retinex filter [Choi et al., 2007] to account for illumination variations. The system was designed such that an experimenter would collect calibration points offline (typically 200), to do the training of the SVR model and the offline processing the videos. This methodology was succesfully applied to children and adults, with a decrease in accuracy for children.

Martinez et al. [2012] also relied on a head mounted setup. The authors proposed to use multi-level Histogram of Oriented Gradients (mHoG) [Dalal and Triggs, 2005] as appearance features to train a Support Vector Regression (SVR), or Relevance Vector Regression (RVR) models. The advantage is that HoG can better cope with illumination variations, in contrast to intensity based features.

In spite of their robustness to image resolution (as opposed to feature based methods), appearance based methods suffer from generalization problems. In the following, we will discuss strategies used to provide invariances to head pose variations and to the given subject.

Handling head pose variations

Among the previously described methods, only the approach by [Sugano et al., 2008] addressed variations due to head pose. Nevertheless, this method suffered from a significant increase of gaze estimation errors and from the amount of needed training data.

Recently, this problem gained increased attention. Lu et al. [2011b] proposed to use the same reconstruction methodology as in [Lu et al., 2011a] (based on calibration samples collected for a single pose), and apply it even under head pose variations at test time. To handle the gaze estimation bias caused by the viewpoint mismatch due to head pose, they proposed to apply a correction method based on a gaussian process regression mapping learned from an additional short video including head pose variations. The same authors [Lu et al., 2012] proposed to collect calibration samples corresponding to a single head pose, and a few extra samples collected under different head poses. These extra samples were used to warp the original calibration set to other viewpoints, thus synthesizing the eye appearance needed to match head pose at test time. Gaze would then be inferred from the sparse linear reconstruction

of the view-dependent samples. Altogether, however, these methods still require additional training data and complex models to capture head-pose related appearance variations.

In another direction, Funes Mora and Odobez [2012] proposed an eye appearance rectification step based on depth data. This method is covered in detail in Chapter 5. Very recently, a similar strategy was used by Egger et al. [2014], but the authors relied on a 3D face model fitted to the 2D image, rather than depth measurements.

Appearance variations across people

The problem of appearance variations across people, or person invariance, has not received much attention. Only Noris et al. [2008] addressed this problem prior to the development of this thesis. In their work, an SVR or GPR model was used to map the eye images, retrieved from a head mounted camera, into the $\mathbf{p}_{PoR}$ within a first person view egocentric video. These models were trained offline from a gaze annotated database of 33 adults. Acceptable results of 2.34° were obtained in this database, but it was not evaluated on children, which were the end target subjects. Interestingly, their following work ([Noris et al., 2010], described previously) instead proposed for an examiner to collect the calibration points offline, to process the small children data.

In this thesis, we therefore addressed the person invariance problem when using remote sensing. Our contributions [Funes Mora and Odobez, 2013, 2015] will be covered in detail in Chapter 5. For the sake of completeness we will here mention a few very recent works addressing this problem as well.

Schneider et al. [2014] proposed a dimensionality reduction method. By defining each subject in the dataset as a separate class, the method was designed to maximize the intra-class distance while minimizing the inter-class distance of gaze synchronized samples in the resulting lower dimensional space. The authors also made extensive experiments on the Columbia Gaze Data Set [Smith et al., 2013], showing interesting combinations between features and regression techniques. However, the used data was of very high quality, and the feature extraction was based on the accurate localization of eye corners.

Sugano et al. [2014] proposed to train random forests for regressing the gaze parameters from both the eye appearance and the head pose parameters jointly. To augment the training set in terms of variations due to head poses, they used a multi view camera array during the data collection, such that pose dependent samples could be later synthesized using 3D multi view reconstruction. By aggregating the data from multiple subjects, the model was also made person invariant. Although promising, evaluations were conducted assuming a perfectly estimated head pose and eyes localization, based on manual annotations of eye corners.

Finally, within this context, it is important to mention that good eye image localization (*alignment*) is a crucial step when it comes to training and testing person invariant appearance based gaze estimation models. Inconsistent eye image extraction across subjects directly

impacts the computation of the eye feature vector, which will then have an impact on the regression model, both at the training and testing phases.

This problem, however, has not received much attention. Mostly because, when working with user and single session dependent models (or with little pose variations), a single cropping is assumed which usually remains consistent for all data points. Otherwise, this step is normally assumed to be done manually [Lu et al., 2011a, Martinez et al., 2012, Sugano et al., 2014], based on a supervised regression method [Noris et al., 2008], or using an automatic eye corner detection methods (for example, the Omron software [Smith et al., 2013, Schneider et al., 2014]) but this normally requires high resolution images.

2.3 Conclusions

Provided the previous coverage of prior works, we will now discuss and summarize their main characteristics and limitations. Motivated by this discussion, we will introduce the research directions of this thesis, and briefly describe how our contributions address the limitations of prior works. For this discussion it is important to consider the temporal context of each contribution.

Head pose estimation

As discussed in Section 2.1, prior works have shown that tracking methods deliver the highest head pose estimation accuracy. Furthermore, model based methods are more robust, accurate and have potential to define a stable and semantically consistent **HCS**. This is crucial for the task of gaze estimation, in order to circumvent the eye localization problem through the definition of the eyeball position as a fixed point with respect to the **HCS**.

The addition of the depth modality, through the introduction of consumer RGB-D sensors, further boosted the accuracy of head pose tracking methods. In particular, impressive head pose estimation accuracy was obtained using the ICP algorithm with the support of person-specific 3D face models [Weise et al., 2011]. Moreover, with the use of robust estimators, it becomes possible to address large occlusions and extreme head poses. In addition, person-specific face models implicitly define the required **HCS**. Nevertheless, in the work of Weise et al. [2011], the person-specific face model was built from non-rigid registration techniques which required data collection through a specific user cooperative protocol.

Motivated by these findings, in [Funes Mora and Odobez, 2012] we proposed to exploit consumer RGB-D sensors for gaze estimation. Aiming to avoid the need for explicit participant cooperation, instead of a particular data collection protocol, we proposed to rely on 3DMMs for the creation of person specific face models in an offline phase. These generative models are able to span a large set of facial shapes and to deliver semantically consistent instances. This resulted in an accurate head pose estimation algorithm, capable of addressing extreme head

poses and to deliver time consistent estimates of the eye location. In this thesis we further extended the framework proposed in [Funes Mora and Odobez, 2012] such that the person specific model is built online during tracking. This is described in detail in Chapter 3.

Gaze estimation

In the context of gaze estimation (cf. Sec. 2.2), two broad categories have been identified and discussed: geometric based and appearance based methods. We can conclude that geometric based methods can be very accurate but rely strongly on the extraction of local eye features. This procedure thus require high resolution and high contrast images, normally obtained through IR setups. To remove the need for specialized and costly hardware, and to increase the range of operation, in terms of user-sensor distance, amount of head orientations, and amount of gaze directions, it becomes necessary to find a solution which does not require to extract local features.

Appearance based methods were proposed as such solution, as they rely on regression techniques. Nevertheless, since as early as the work of Baluja and Pomerleau [1994], it was identified that generalization to variations due to head pose will be an important challenge for these approaches to address. Therefore, most works focused on a single viewpoint relative to the head, either by requiring for the participant to maintain a static head pose (e.g., using a chin-rest), or by wearing head mounted cameras. This clearly demands full cooperation from the participant and significantly limits the set of plausible applications. Recents works started to address the problem of head pose variations. Nevertheless, additional calibration data collected under head poses variations was necessary, which is still limiting.

Furthermore, in appearance based methods, the problem of generalization extends to more variables than the head pose. Other elements need to be addressed, of which the variations in the eye appearance across users is a prominent case. To acquire *person invariance* is of utmost importance in non-cooperative scenarios. Notice all previous works relied on conditions in which the regression models were trained and evaluated on the same subject (except [Noris et al., 2008]). In addition, experimental validations were always conducted using a private dataset. This makes difficult to properly characterize previous methods objectively, in terms of their merits and limitations when contrasted to other approaches.

In summary, gaze estimation solutions which avoid local features tracking, and that are robust to variations in terms of head pose and user appearance are needed. Therefore, in this thesis, we took the research directions which we describe as follows (in the order of their corresponding chapters):

- **EYEDIAP database [Chapter 4]**. The main reason why prior works are evaluated mostly in private datasets is the lack of public benchmarks to train and evaluate gaze estimation methods. Therefore we collected and made publicly available the EYEDIAP database, which is rich in variations in terms of users, scenarios, head pose and ambient conditions.

- **Appearance based methods [Chapter 5]**. Due to the limitation of geometric based methods to address low resolution sensing conditions, we decided to first investigate on the appearance based paradigm. We directly address the challenges of head pose and user invariance, which, as described previously, have been an important limitation for prior works. We propose methods to correct the eye image appearance variation due to head pose, and therefore allowing for head pose invariant appearance based gaze estimation. We also investigate on the creation of person invariant gaze models by relying on the EYEDIAP database. In addition, we propose an eye image alignment method which differs from the few strategies found in the literature and which improve the accuracy of person invariant gaze estimation models.
- **Geometric generative gaze estimation [Chapter 6]**. To address the limitations of appearance based methods in terms of generalization and, furthermore, in terms of adaptation, we propose an alternative methodology called *geometric generative gaze estimation* (G^3E). This is a new paradigm to the gaze estimation problem which, similarly to appearance based methods, does not require to track local features, however, and in contrast, it is based on a geometric modelling of the eyeball and image formation process. This method has important advantages with respect to appearance based methods and standard geometric based methods.
- **Gaze coding in natural dyadic and group interactions [Chapter 7]**. We investigate on the problem of gaze coding in real and challenging human-human interaction scenarios. This application relies on the head pose and user invariance of the proposed contributions, and benefits from the overall 3D gaze tracking methodology. Therefore, this application further validates the proposed methods and is an example scenario on how to exploit the contributions of this thesis.

3 3D Head Pose Tracking

In this chapter we describe in detail the 3D head pose estimation method developed in this thesis. As motivated in Section 2.3, we exploit RGB-D data from consumer sensors as input. The head pose tracker is mainly based on applying the iterative closests points (ICP) algorithm to register a person specific face model to the depth data input. Aiming to minimize the need for user cooperation, we rely on 3D Morphable Models (3DMM) to build the person specific models in either an offline or online fashion.

In Section 3.1, we cover background information; elements such as the 3DMM and the basic ICP algorithm are described in detail. Then we present the two main head pose tracking methodologies we developed. The first one applies the ICP based head pose tracking assuming the person specific face model is available a priori. To obtain this model, we propose an *offline* and supervised strategy to fit a 3DMM to RGB-D data samples. Therefore, this first methodology will be described in two sections: Section 3.2 describes the proposed offline 3DMM fitting, whereas the ICP based head pose tracking is described in detail in Section 3.3. In the second approach, described in Section 3.4, we extend the previous methodology to both fit the 3DMM and track the subject's face *online*, jointly. In Section 3.5 we present experiments to validate the methods. Finally, we conclude the Chapter with a discussion in Section 3.6.

3.1 Background

In this section we cover the background elements needed to describe the proposed methods, in particular, the definition of 3D morphable models and the iterative closest points algorithm.

3.1.1 3D Morphable Models

Let $\mathbf{x}$ represent an object's shape as a column vector which contains the N_v 3D vertices of said object, i.e., $\mathbf{x} := (x_1, y_1, z_1, x_2, y_2, z_2, \ldots, x_{N_v}, y_{N_v}, z_{N_v})^\top$. Assume $\mathbf{x}$ belongs to a class of objects, in which each instance differ through variations in shape, but share the same structure (or topology). 3D Morphable Models, as a direct extension of ASM and AAM, assume the class of

objects lie within a linear subspace of dimensionality $N_{\mathbf{M}}$. Therefore, a 3DMM represents the object class as a generative model, from which an instance $\mathbf{x} \in \mathbb{R}^{3N_v}$ is retrieved as a function of a vector $\alpha \in \mathbb{R}^{N_{\mathbf{M}}}$, as follows:

$$\mathbf{x}(\alpha) = \mu + \mathbf{M}\alpha, \tag{3.1}$$

where $\mu \in \mathbb{R}^{3N_v}$ is the mean shape of the class of objects, and $\mathbf{M} \in \mathbb{R}^{3N_v \times N_{\mathbf{M}}}$ contains a set of deformations from the mean shape. Normally $N_{\mathbf{M}} << 3N_v$, meaning that α is a much lower dimensional representation for $\mathbf{x}$, and $\mathbf{M}$ embeds the set of possible shape deformations for the given class.

This process is applied to the object's geometry, as a point cloud in the 3D space. Nevertheless, a topology $\mathcal{T}$ can also be defined to extend the point cloud into a 3D mesh. Here, $\mathcal{T} := \{f_m\}_{m=1}^{N_{\mathcal{T}}}$ is a set of facets, where each is defined by triplets of vertex indices $f_m = (i, j, k)$, meaning the facet m defines a triangle composed by the 3D vertices i, j and k.

Similarly to AAM, it is possible to represent the class *texture* as a linear subspace. In such case, the texture is normally defined as a shape free 2D image with a predefined correspondence to the 3DMM vertices (as AAM, see Figure 2.3), or, an intensity value can be assigned per vertex. In either case, a texture's instance $\tau \in \mathbb{R}^{N_\tau}$ (assuming N_τ single channel pixels) is obtained as:

$$\tau(\alpha_\tau) = \mu_\tau + \mathbf{M}_\tau \alpha_\tau, \tag{3.2}$$

where $\alpha_\tau \in \mathbb{R}^{N_{\mathbf{M}_\tau}}$ define the model coefficients associated to the texture's instance τ; $\mu_\tau \in \mathbb{R}^{N_\tau}$ is the mean texture and $\mathbf{M}_\tau \in \mathbb{R}^{N_\tau \times N_{\mathbf{M}_\tau}}$ represents the variation basis.

Previous works have addressed the fitting of a 3DMM into 2D images, that is, finding the parameters α, α_τ and a *rigid* transform which best describes the data. As the 3DMM encodes a generative process and can synthesize instances, the cost is formulated as the discrepancy between the image data and the model's synthetic instance. This is equivalent to the AAM cost, with the addition of a projective transform referring the 3D geometry to the 2D image coordinates. For details see, e.g., [Vetter and Blanz, 1998] and the PhD thesis by Reinhard [2009]. Nevertheless, in this work, we are mainly interested in shape based registration.

Expressions modelling

Although the main focus in this thesis is on rigid face tracking, as needed for the gaze tracking task, a 3DMM can be defined to model both identity and expressions related deformations. The most common strategy is to assume both elements lie within independent linear subspaces

which can be linearly combined as shown in Equation 3.3.

$$\mathbf{x}(\alpha) = \mu + \mathbf{M}\alpha + \mathbf{E}\beta, \tag{3.3}$$

where we assume $\mathbf{M}$ and α are only due to identity related deformations. $\mathbf{E} \in \mathbb{R}^{3N_v \times N_\beta}$ represents a facial expression deformation basis. The facial expressions are then parametrized by the coefficients $\beta \in \mathbb{R}^{N_\beta}$.

3.1.2 Iterative Closest Points

The iterative closest points algorithm (ICP) is used for the rigid *registration* of 3D meshes. It was initially proposed by Chen and Medioni [1991] and Besl and McKay [1992]. ICP finds the rotation $\mathbf{R} \in \mathbb{R}^{3\times3}$ and translation $\mathbf{t} \in \mathbb{R}^3$ which minimises the surface error distance, as measured by its vertices.

Generic ICP formulation

Let us assume the objective is to align a *template* object $\mathbf{V}$ with the *target* data $\mathbf{U}$. Both objects (or point clouds) are defined as $\mathbf{V} := \left\{\{\mathbf{v}_k\}_{k=1}^{N_\mathbf{V}}, V\right\}$ and $\mathbf{U} := \left\{\{\mathbf{u}_k\}_{k=1}^{N_\mathbf{U}}, U\right\}$, i.e., they are composed of $N_\mathbf{V}$ and $N_\mathbf{U}$ 3D points respectively, together with the optional sets V and U. These sets represent an optional augmentation of the point clouds, e.g., including a topology, normals, color, etc. In general, an appropriate cost function to evaluate the alignment of the point clouds $\mathbf{V}$ and $\mathbf{U}$, is the following:

$$E_{ICP}(\mathbf{R},\mathbf{t}) = \sum_{k=1}^{N_\mathbf{V}} dist\left(\mathbf{R}\mathbf{v}_k + \mathbf{t}, \mathbf{u}_{C(k)}\right)^2, \tag{3.4}$$

where $dist$ defines a distance function between two points (e.g., euclidean distance) and C is a mapping which returns the index of the *correspondence* point in the target cloud for the point k in the template. In other words, if T represents an operator applying a rigid transformation, then Eq. 3.4 reaches a minimum whenever $T(\mathbf{V};\mathbf{R},\mathbf{t})$ and $\mathbf{U}$ are well aligned.

The challenge with this formulation is that, both C and $\{\mathbf{R},\mathbf{t}\}$ are not known. Nevertheless, ICP is based on the following observation: if the correspondences between $\mathbf{V}$ and $\mathbf{U}$ were known (i.e., the mapping C), then the optimal rigid transform $\{\mathbf{R},\mathbf{t}\}$ could be computed from pairs of 3D correspondences based on procrustes analysis[1]. Alternatively, if the optimal rigid transform was known, it should be possible to find the optimal correspondence mapping. Therefore,

[1] Excluding the degenerative case of colinear sets of points

Algorithm 1 : Generic ICP algorithm.

1: Initialize $\{\mathbf{R}, \mathbf{t}\}$
2: **while** not converged **do**
3: **Step A**. Set C by searching correspondences of $T(\mathbf{V}; \mathbf{R}, \mathbf{t})$ in $\mathbf{U}$
4: **Step B**. Use C to solve:

$$\{\hat{\mathbf{R}}, \hat{\mathbf{t}}\} = \arg\min_{\mathbf{R},\mathbf{t}} E_{ICP}(\mathbf{R}, \mathbf{t}) \tag{3.5}$$

5: Set $\mathbf{R} \leftarrow \hat{\mathbf{R}}$, $\mathbf{t} \leftarrow \hat{\mathbf{t}}$ ▷ Update
6: **end while**
7: Return $\{\mathbf{R}, \mathbf{t}\}$

the ICP algorithm, which is summarized in Algorithm 1, proposes an iterative approach, alternating between the rigid transform computation and the search for correspondences.

Many variants of this basic formulation have been proposed in the literature. They differ on the correspondence search step, on the used distance function, on whether robust estimators are employed to handle noise or missing data, on the outliers handling strategy, on whether the rigid transform computation is incremental or absolute, etc. For an overview, please see the survey by Rusinkiewicz and Levoy [2001]. Furthermore, non rigid formulations have been proposed to register free meshes, e.g., [Amberg et al., 2007]. The main limitation of ICP is that it requires a good initialization, as it can otherwise converge to local minima.

In the following, we will describe elements of the ICP implementation we used and that are common for the rest of the chapter.

Correspondence search

There are different strategies to find the correspondence in the target mesh $\mathbf{U}$, for a point $\mathbf{v}_k$, with a normal $\mathbf{n}_k$, in the transformed template $T(\mathbf{V}; \mathbf{R}, \mathbf{t})$. The most common method is the *closest point* search, i.e., we assign the point with the minimal euclidean distance to $\mathbf{v}_k$. *Normal shooting*, instead, assigns the closest point in $\mathbf{V}$, but found along the direction of $\mathbf{n}_k$. Alternatively, the *point to projection* search projects $\mathbf{v}_k$ into the depth data image, retrieving a depth pixel, which is then backprojected to create the 3D correspondence point. However, this requires for a depth map and the camera calibration parameters to be available.

As found empirically by Rusinkiewicz and Levoy [2001], the normal shooting search is more robust to noise in comparison to the other methods. However, its implementation can be computationally costly. In contrast, the point to projection search is fast and its complexity is independent of the number of target data points. Therefore, in this thesis, we used the method proposed by Park and Subbarao [2003]. Their approach formulates the normal shooting correspondence search as a sequence of point to projection search steps. This approach provides a good trade-off between accuracy and computational efficiency.

Incremental ICP

Instead of solving Eq. 3.5, to find the optimal $\{\mathbf{R},\mathbf{t}\}$ global alignment, we reformulate the problem in terms of an incremental rigid transform $\{\delta\mathbf{R},\delta\mathbf{t}\}$ as follows:

$$\{\hat{\delta\mathbf{R}},\hat{\delta\mathbf{t}}\} = \arg\min_{\delta\mathbf{R},\delta\mathbf{t}} \sum_{k=1}^{N_V} dist\left(\delta\mathbf{R}\left(\mathbf{R}\mathbf{v}_k+\mathbf{t}\right)+\delta\mathbf{t},\mathbf{u}_{C(k)}\right)^2, \tag{3.6}$$

Whereas the rigid transform update becomes:

$$\mathbf{R} \leftarrow \hat{\delta\mathbf{R}}\mathbf{R} \tag{3.7}$$

$$\mathbf{t} \leftarrow \hat{\delta\mathbf{R}}\mathbf{t}+\hat{\delta\mathbf{t}} \tag{3.8}$$

By formulating the algorithm in terms of increments, we can approximate $\delta\mathbf{R}$ by a linear form with respect to the euler angles, allowing to solve Eq. 3.6 using linear least squares [Morency and Darrell, 2002, Lowe, 2004]. This approximation relies on the euler angles being close to zero. Nevertheless notice that this approximation becomes accurate near the optimal rigid transform ($\{\mathbf{R},\mathbf{t}\}$), where the change in euler angles tend to zero.

Robust estimators

To gain robustness against outliers and missing data, the ICP method can be further enhanced based on robust estimators and heuristics, by extending Eq. 3.4 as follows:

$$E_{ICP}(\mathbf{R},\mathbf{t}) = \sum_{k=1}^{N_V} w_k dist\left(\mathbf{R}\mathbf{v}_k+\mathbf{t},\mathbf{u}_{C(k)}\right)^2, \tag{3.9}$$

The per-vertex weight $w_k \in [0,1]$ is used to filter out bad correspondences, and it is obtained as follows. It is reestimated at each ICP iteration, between step A and B (see Algorithm 1). It is either set inversely proportional to the euclidean distance between two correspondences, or it is directly set to zero if any of the following situations are true:

1. **Maximum distance**. The euclidean distance between the correspondences is larger than 5cm. This is a strong indicator of a bad correspondence, specially if we assume the template and target mesh do not differ much in terms of position.

2. **Normals compatibility**. The angle between the surface norms (between the template and target), at the correspondences position, is larger than a threshold (typically 45°).

Again, this is normally indicative of bad matches, as the surfaces should be facing roughly a similar direction.

3. **Mesh border**. The correspondence for a point k is in the border of the target mesh. This is particularly useful to handle incomplete data. This is recommended by Rusinkiewicz and Levoy [2001]. As to filter such correspondences can be computationally costly, this verification can be left as optional. In this thesis, we discarded this test whenever the $dist$ function is defined as a point-to-plane cost (cf. Section 3.3.1) which we found more robust to wrong correspondences along the mesh border.

3.2 Person-specific face model learning

In this section we describe the algorithm proposed to create the person specific face model from the 3DMM fitting to RGB-D data. This is formulated as an *offline* step, to be done prior to the head pose tracking. We do not assume the subject has to undergo a particular protocol for data collection.

3.2.1 Multiple instance 3DMM fitting

The algorithm we propose here is part of the gaze tracking framework described in [Funes Mora and Odobez, 2012]. It is based on the method proposed by Amberg et al. [2008], originally intended for the face recognition task, under facial expressions, from 3D high quality range scans. The authors proposed to fit a 3DMM, linearly combining both expressions and identity related deformations (as shown in Eq. 3.3), to a 3D scan data using a non-rigid ICP formulation. In their approach, the fitting algorithm simply combined both sources of deformations into a single deformation matrix and deformation parameters (as in Eq. 3.1).

We here propose to extend the method of Amberg et al. [2008] by fitting the 3DMM to a collection of J sample meshes $\{\mathbf{U}_j\}_{j=1}^{J}$ (RGB-D frames containing the subject's face), and to constrain the fitting based on landmarks positions. The main motivation is that RGB-D data, specially from consumer sensors, has high levels of noise, and significant portions of depth data are missing. We will here assume the 3DMM deformation basis spans only identity related shape variations. At the end of this section we will discuss facial expressions handling.

More formally, to learn the person specific 3D facial mesh, we find the 3DMM instance that best fit a set of J samples (RGB-D images) of the subject, by solving the following optimization problem:

$$\hat{\mathbf{X}} = \arg\min_{\mathbf{X}} \left(\lambda E_s(\mathbf{X}) + \sum_{j=1}^{J} E_d^j(\mathbf{X}) + \gamma E_l^j(\mathbf{X}) \right), \tag{3.10}$$

where the parameters $\mathbf{X} := \{\alpha, \mathbf{R}_1, \mathbf{t}_1, \ldots, \mathbf{R}_J, \mathbf{t}_J\}$ to optimize are the 3DMM coefficients α, and those of a rigid transformation (defined by a rotation $\mathbf{R}_j$ and translation $\mathbf{t}_j$) for each RGB-D sample j. Note, the key idea is that the α parameters are shared between the instances, thus all samples contribute to their estimation. The different cost terms are defined as follows:

$$E_d{}^j(\mathbf{X}) := \sum_{i=1}^{N_v} \|F_d{}_i^j(\mathbf{X})\|^2; \qquad E_l{}^j(\mathbf{X}) := \sum_{i \in L} \|F_l{}_i^j(\mathbf{X})\|^2; \quad and \quad E_s(\mathbf{X}) := \|F_s(\mathbf{X})\|^2, \tag{3.11}$$

where

$$F_d{}_i^j(\mathbf{X}) = w_i^{0.5}(\mathbf{R}_j(\mu_i + \mathbf{M}_i\alpha) + \mathbf{t}_j - \mathbf{u}_{C_j(i)}) \tag{3.12}$$

$$F_l{}_i^j(\mathbf{X}) = \mathbf{R}_j(\mu_i + \mathbf{M}_i\alpha) + \mathbf{t}_j - \mathbf{l}_i^j \tag{3.13}$$

$$F_s(\mathbf{X}) = \alpha \tag{3.14}$$

Notice that μ_i and $\mathbf{M}_i$ represent the 3 rows corresponding to the vertex i in μ and $\mathbf{M}$. The data term E_d represents the cumulative distance of each deformed and rigidly transformed vertex i of the 3DMM to its closest point in the data (cf. Eq. 3.12), represented by $\mathbf{u}_{C_j(i)}$.

The term E_l is similar to the E_d cost, but applies to a set of N_L landmarks points (which form a subset L of the 3DMM vertices). Their position $\mathbf{l}_i^j$ is assumed to be *manually* annotated in the data. This term fosters a semantic fitting of the 3DMM (eye corners, eyebrows, mouth corners, etc.) which, due to depth noise in the data, could be otherwise poorly localized. Finally, the regularization term E_s foster the estimation of small values for α. This term is weighted by the *stiffness* parameter λ, controlling how much the instance mesh can deform. The solution to Equation 3.10 is found through a non-rigid ICP fitting procedure, which is explained in the following Section.

3.2.2 Non-rigid ICP fitting

Define $\mathbf{V}(\alpha)$ as the mesh generated from the 3DMM, given the α parameters. Therefore, to find the optimal parameters $\hat{\mathbf{X}}$, we fit the 3DMM to data using the non-rigid ICP formulation described in Algorithm 2.

This algorithm systematically reduce the stiffness value, allowing for larger deformations as the correspondences are more accurate. This is a common strategy in non-rigid ICP methods, e.g., [Amberg et al., 2007].

The initialization ($\mathbf{X}^0 := \{\alpha^0, \mathbf{R}_1^0, \mathbf{t}_1^0, \ldots, \mathbf{R}_J^0, \mathbf{t}_J^0\}$) corresponds to that of the mean facial shape ($\alpha^0 = \mathbf{0}$) and its -per sample j- rigid transformation which minimizes the landmarks term $E_l{}^j$ alone, assuming $\alpha = \alpha^0$.

Algorithm 2 : 3DMM fitting optimization algorithm.

Initialize $\mathbf{X}$ as $\mathbf{X}^0$.
Set $k = 0$
for each stiffness value $\lambda^n \in \Lambda$, where $\Lambda := \{\lambda^n | \lambda^n > \lambda^{n+1}\}$, **do**
  while $\|\mathbf{X}^k - \mathbf{X}^{k-1}\| > \epsilon$ **do**
    for each instance j **do**
      Extract $\{\alpha^k, \mathbf{R}_j^k, \mathbf{t}_j^k\}$ from $\mathbf{X}^k$
      Compute C_j based on the correspondences of $T(\mathbf{V}(\alpha^k); \mathbf{R}_j^k, \mathbf{t}_j^k)$ in $\mathbf{U}_j$
      Compute the robust weights $\{w_i\}_j$
    end for
    $k \leftarrow k+1$
    Compute $\mathbf{X}^k$ as the solution of Eq. 3.10 using $\{C_j\}$,λ^n and $\{w\}$ ▷ Gauss-Newton
  end while
end for
Set $\hat{\mathbf{X}} \leftarrow \mathbf{X}^k$
Return $\hat{\mathbf{X}}$

Gauss-Newton optimization

To find $\hat{\mathbf{X}}$ as the solution of Eq. 3.10 based on fixed $\{C_j\}$,λ^n and $\{w\}$, we followed the formulation of Amberg et al. [2008], i.e., the rigid transform is inverted (applied to the target data), as this allows to separate the α coefficients from the rigid transform parameters. Then, Gauss-Newton is used to optimize the cost function as a pseudo-Newton gradient descent algorithm.

The main difference here is the definition of the Jacobian. Amberg et al. [2008] proposed to precompute some elements of the Hessian, prior to efficiently solve each of the gradient descend steps. We found this needed to be revised. In particular, this is due to the need to recompute the robust weights each time the correspondences are found. Our Jacobian formulation is also different, due to the joint fitting to multiple instances, and additional terms in the cost function. Let us first define the following terms:

$$F_d{}^j(\mathbf{X}) := \begin{bmatrix} F_d{}_1^j(\mathbf{X}) \\ F_d{}_2^j(\mathbf{X}) \\ \vdots \\ F_d{}_{N_v}^j(\mathbf{X}) \end{bmatrix}; \qquad and \qquad F_l{}^j(\mathbf{X}) := \begin{bmatrix} F_l{}_1^j(\mathbf{X}) \\ F_l{}_2^j(\mathbf{X}) \\ \vdots \\ F_l{}_{N_L}^j(\mathbf{X}) \end{bmatrix}, \tag{3.15}$$

where the per point and per landmark terms are given in Eq. 3.12 and Eq. 3.13. Then the

Jacobian, in general, takes the form shown in Equation 3.16:

$$\mathbf{J}(\mathbf{X}) = \begin{bmatrix} \frac{\partial {F_d}^1}{\partial \alpha} & \frac{\partial {F_d}^1}{\partial \mathbf{t}_1} & \mathbf{0} & \dots & \mathbf{0} & \frac{\partial {F_d}^1}{\partial \mathbf{R}_1} & \mathbf{0} & \dots & \mathbf{0} \\ \frac{\partial {F_l}^1}{\partial \alpha} & \frac{\partial {F_l}^1}{\partial \mathbf{t}_1} & \mathbf{0} & \dots & \mathbf{0} & \frac{\partial {F_l}^1}{\partial \mathbf{R}_1} & \mathbf{0} & \dots & \mathbf{0} \\ \frac{\partial {F_d}^2}{\partial \alpha} & \mathbf{0} & \frac{\partial {F_d}^2}{\partial \mathbf{t}_2} & \dots & \mathbf{0} & \mathbf{0} & \frac{\partial {F_d}^2}{\partial \mathbf{R}_2} & \dots & \mathbf{0} \\ \frac{\partial {F_l}^2}{\partial \alpha} & \mathbf{0} & \frac{\partial {F_l}^2}{\partial \mathbf{t}_2} & \dots & \mathbf{0} & \mathbf{0} & \frac{\partial {F_l}^2}{\partial \mathbf{R}_2} & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial {F_d}^J}{\partial \alpha} & \mathbf{0} & \mathbf{0} & \dots & \frac{\partial {F_d}^J}{\partial \mathbf{t}_J} & \mathbf{0} & \mathbf{0} & \dots & \frac{\partial {F_d}^J}{\partial \mathbf{R}_J} \\ \frac{\partial {F_l}^J}{\partial \alpha} & \mathbf{0} & \mathbf{0} & \dots & \frac{\partial {F_l}^J}{\partial \mathbf{t}_J} & \mathbf{0} & \mathbf{0} & \dots & \frac{\partial {F_l}^J}{\partial \mathbf{R}_J} \\ \frac{\partial F_s}{\partial \alpha} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \end{bmatrix}, \tag{3.16}$$

where the entry **0** represents a matrix full of zeros; its size can be determined from its position within the Jacobian matrix. To avoid clutter, we omit the dependency to **X**, and do not show the weights of the different terms of the cost function. Also, we show the derivatives with respect to the forward rigid transform but, in practice, it is computed with respect to the inverse rigid transformation [Amberg et al., 2008]. Finally, the derivative with respect to the -inverse- rotation matrix is, in fact, computed with respect to its euler angles.

Note that the Jacobian matrix's size is $(3JN_v + 3JN_L) \times (N_{\mathbf{M}} + 6J)$. This assumes the number of landmarks is constant, but it may change per sample, according to which landmarks are visible or occluded. In addition, the amount of zeros within **J** is large, and some elements remain unchanged during the gradient descent, therefore, the Jacobian, the Gauss Newton approximation of the Hessian, and the Newton step may be computed efficiently.

3.2.3 Facial expressions handling

For the work developed in this thesis we ignored facial expressions, due to practical reasons (see Section 3.5). The previous formulation therefore assumed the deformation parameters α only model shape variations due to identity. Nevertheless, the proposed framework can no longer be extended in a straightforward way, by assuming α includes both the facial expressions and the identity related coefficients, as in the case of Amberg et al. [2008]. This is due to the fact that, each data sample, may have a different facial expression. We here discuss the extension needed to address facial expressions.

Assuming the 3DMM is defined as in Eq. 3.3, the terms of Equations 3.12, and 3.13 are redefined as:

$$F_{d_i}{}^j(\mathbf{X}) = w_i^{0.5}(\mathbf{R}_j(\mu_i + \mathbf{M}_i\alpha + \mathbf{E}_i\beta_j) + \mathbf{t}_j - \mathbf{u}_{C_j(i)}) \tag{3.17}$$

$$F_{l_i}{}^j(\mathbf{X}) = \mathbf{R}_j(\mu_i + \mathbf{M}_i\alpha + \mathbf{E}_i\beta_j) + \mathbf{t}_j - \mathbf{l}_i^j \tag{3.18}$$

where the α coefficients are shared between the J samples, but the β parameters (linked to facial expressions) are modelled separately, per sample j. $\mathbf{E}_i$ represents the 3 rows of $\mathbf{E}$ corresponding to the point i (as is the case for $\mathbf{M}_i$ and μ_i). The stiffness term in the cost function is also redefined as:

$$E_s(\mathbf{X}) := \|F_s(\mathbf{X})\|^2 + \sum_{j=1}^{J} \|F_\beta{}^j(\mathbf{X})\|^2 \tag{3.19}$$

where $F_\beta{}^j(\mathbf{X}) = \beta_j$. Therefore, we also regularize with respect to the facial expressions. The non-rigid ICP approach, described in Algorithm 2, is used in the same way as with the expression-less formulation, but it is extended to consider $\mathbf{X}$ now includes $\{\beta_j\}$.

If we take these modifications into consideration, and assuming $\mathbf{J}(\mathbf{X})$ is defined as in Equation 3.16, then the Jacobian considering expressions gets augmented as shown in Equation 3.20.

$$\mathbf{J_E}(\mathbf{X}) = \left[\begin{array}{c|cccc} & \frac{\partial F_d{}^1}{\partial \beta_1} & \mathbf{0} & \dots & \mathbf{0} \\ & \frac{\partial F_l{}^1}{\partial \beta_1} & \mathbf{0} & \dots & \mathbf{0} \\ & \mathbf{0} & \frac{\partial F_d{}^2}{\partial \beta_2} & \dots & \mathbf{0} \\ \mathbf{J}(\mathbf{X}) & \mathbf{0} & \frac{\partial F_l{}^2}{\partial \beta_2} & \dots & \mathbf{0} \\ & \vdots & \vdots & \ddots & \vdots \\ & \mathbf{0} & \mathbf{0} & \dots & \frac{\partial F_d{}^J}{\partial \beta_J} \\ & \mathbf{0} & \mathbf{0} & \dots & \frac{\partial F_l{}^J}{\partial \beta_J} \\ & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \hline & \frac{\partial F_\beta{}^1}{\partial \beta_1} & \mathbf{0} & \dots & \mathbf{0} \\ & \mathbf{0} & \frac{\partial F_\beta{}^2}{\partial \beta_2} & \dots & \mathbf{0} \\ \mathbf{0} & \vdots & \vdots & \ddots & \vdots \\ & \mathbf{0} & \mathbf{0} & \mathbf{0} & \frac{\partial F_\beta{}^J}{\partial \beta_J} \end{array}\right] \tag{3.20}$$

3.3 Model-based head pose tracking

In this section the head pose tracking algorithm formulation is described. We assume a person specific 3D face model (*template*) was obtained offline, by using the 3DMM fitting algorithm described in Section 3.2. In addition, we assume the RGB-D input data has been processed to generate a 3D mesh[2], as shown in Fig. 1.4b. This section follows closely the method used by Weise et al. [2011].

3.3.1 ICP based head pose tracking

The task at hand consists on obtaining the rigid transform $\{\mathbf{R}_t, \mathbf{t}_t\}$ which best aligns the template $\mathbf{V}$ to the input data $\mathbf{U}_t$ at time t. This formulation is shown in the following equation,

$$\{\hat{\mathbf{R}}_t, \hat{\mathbf{t}}_t\} = \arg\min_{\{\mathbf{R}_t, \mathbf{t}_t\}} \sum_{i \in \mathbf{V}_U} w_i \left((\mathbf{R}_t \mathbf{n}_i)^\top \left(\mathbf{R}_t \mathbf{v}_i + \mathbf{t}_t - \mathbf{u}^t_{C(i)} \right) \right)^2, \tag{3.21}$$

where we can observe new elements when comparing to Eq. 3.4. The $dist$ function has been replaced here by the *point-to-plane* cost, where $\mathbf{n}_i$ is the template normal at point $\mathbf{v}_i$, expressed in the reference **HCS**. Empirically, it was found that this formulation converges faster and is more robust to poor initialization, at least in comparison to a *point-to-point* cost [Rusinkiewicz and Levoy, 2001]. These characteristics are valuable in a tracking framework.

During tracking (assuming the system has been successfully initialized), the ICP algorithm at time t will start from the rigid transform obtained at time $t-1$. The main assumption is that the head pose does not changes "much" between frames $t-1$ and t. Neverthless, this depends on the frame rate and the amount of head pose variations, which may be scenario dependent.

With respect to its implementation, the above optimization follows closely the elements described in Section 3.1.2. Meaning, we use the efficient normal shooting correspondence search, the robust weights w_i are defined accordingly, and the implementation solves for increments on the rigid transform. When solving for $\delta\mathbf{R}$ (increment), the normals factor of the incremental formulation of Eq. 3.21 is further approximated as follows:

$$\delta\mathbf{R}\mathbf{R}\mathbf{n}_i \approx \mathbf{R}\mathbf{n}_i \tag{3.22}$$

This is valid as $\delta\mathbf{R} \approx \mathbf{I}_3$ for small angles[3]. The problem thus remains linear with respect to the -incremental- euler angles.

[2] This requires for the RGB-D sensor's extrinsic and intrinsic calibration parameters to be known.

[3] $\mathbf{I}_3$ is the 3×3 identity matrix

Figure 3.1: Face model segment used for the rigid head pose tracking. 3D face model from [Paysan et al., 2009]

Finally, notice that for tracking we only use the set of points in the segment of the face, which are defined by the subset $\mathbf{V}_U$ of $\mathbf{V}$. This set represents a segment of the upper part of the face, as shown in Fig. 3.1. As proposed by Weise et al. [2011], this helps to alleviate the influence of large non rigid deformations around the mouth region, and focus in registering the upper part of the face. Note that this is important for the task of eye localization and gaze estimation.

3.3.2 Initialization

For the overall tracking initialization, we have used two strategies, one based on a frontal face detector and the other on a random forest based regression. Note however that, for all the experiments shown in this thesis, we used the frontal face detector based initialization.

Frontal face detector based initialization

Using the Viola-Jones frontal face detector [Viola and Jones, 2001], available in the OpenCV library, we first detect the face bounding box. We then filter and select the points of the 3D mesh generated from the RGB-D pair which project into the RGB image, inside the face bounding box. Let $\mathbf{t}_{bb}$ represent the median of this set of points, obtained as the median value along each dimension. Therefore, the initial translation is assigned as $\mathbf{t}_0 = \mathbf{t}_{bb} + \mathbf{t}_\mathbf{V}$, where $\mathbf{t}_\mathbf{V}$ is a translation correction defined a priori for a given face model, which takes into account the semantic position of the origin of the **HCS** (which can be, e.g., a point in the neck, the nose or the forehead), with respect to the face surface point which should match $\mathbf{t}_{bb}$. The initial rotation is simply set as $\mathbf{R}_0 = \mathbf{I}_3$ (we assume a frontal face). Once these values are set, the ICP algorithm is used to refine the head pose estimate on this initial frame.

The main problem with this approach is the need for the face to be close to frontal enough to trigger the face detector, and the further assumption of the head to be fully frontal ($\mathbf{R}_0 = \mathbf{I}_3$). The initial rotation can nevertheless be *tailored* according to the scenario (e.g., if the camera is placed at a lower or at the same height as the face), but this does not generalize.

Random forest based initialization

We used the method proposed by Fanelli et al. [2011]. This approach is based on random forest regression using differences of depth patches as features for the split function. It eventually returns both the position of the head and its orientation.

The accuracy of this method is not sufficient to be used for gaze estimation. Nevertheless it is useful to initialize the ICP based head pose tracker. In particular, it can also help to initialize the tracking under non frontal head poses, by providing initial values for ICP. A disadvantage is that the operation range is dependent on the training data used for the regression model.

3.3.3 Failure detection

Once the ICP optimization has been conducted, detecting whether the template and target data are well aligned is a challenging problem on its own. The difficulty lies in the large amount of missing data in the target mesh, due to self occlusion or sensor limitations resulting in missing depth patches at random locations. The latter is common when using consumer RGB-D sensors. Diverse factors may have an influence on this, such as sensing distance, viewing angle, scene's material, illumination conditions, etc. Therefore, the end value of the cost function is non informative. In our implementation we detect failures based on the following heuristics:

Pose change. If the difference in pose between frame $t-1$ and frame t is very large, we assume ICP diverged from the global minimum, into a local minimum. A large change of parameters can be detected by monitoring the euler angles and translation parameters obtained for both frames. Thresholds on their differences can be set to reasonable values, which depend on the frame rate and the expected amount of head pose movements.

Robust weights. We monitor the robust weight value for the points we know should be facing towards the sensor. Notice that the other points can not be detected by the sensor, as they are expected to be self occluded. These two sets of points can be discriminated in the template, based on their normals and the current head pose. Therefore, for the points whose normals are facing towards the sensor, we can assume their robust weight w_i is close to 1. Thus, if $\frac{1}{|S|}\sum_{i\in S} w_i$ is below a threshold, where S denote the set of vertices facing towards the sensor, we assume ICP has failed.

Once a failure has been detected, we reinitialize the tracking, as in Section 3.3.2. Notice that these strategies are prone to errors and further work is needed to improve their robustness. Nevertheless, these criteria have been sufficient for most of the experiments conducted in this thesis.

3.4 Online face model fitting and head pose tracking

A disadvantage of the strategy presented in Section 3.2 is that the person specific face model must be obtained offline, prior to the head pose tracking described in Section 3.3. Even though the cost for this process is acceptable for many applications, both in terms of computation time and the practical cost to collect, and possibly annotate, data instances of the participant's face, it can still be restrictive for applications requiring minimal user cooperation.

Therefore, a solution is to do both tasks at once: to track the head pose while obtaining the person specific face model, and vice versa. Conceptually, this extension is straightforward, as solving Eq. 3.10 frame by frame would lead to the desired result. Notice that Eq. 3.10 solves for both, the 3DMM coefficients α and the pose parameters $\{\mathbf{R}, \mathbf{t}\}$ for each frame.

Whereas in Section 3.2 it was assumed the landmarks were obtained from manual annotations, in an online fitting framework, we assume the landmarks related elements are either discarded ($E_l = 0$), or, landmarks positions are obtained from automatic methods. For completeness, we will assume in this section the landmarks positions are retrieved automatically.

Nevertheless, there are a few reasons why an alternative approach to continuously solving Eq. 3.10 is required:

- The computational cost of the 3DMM fitting, both in terms of processing time and memory, is large. Tracking at a high frame rate (even higher that 1 fps) may not be feasible using consumer hardware.
- Solving Eq. 3.10, frame by frame (if possible) is unnecessary for the goal of obtaining a person specific face model, as consecutive frames deliver similar information.
- The point-to-point constraints used in the data term E_d of Eq. 3.11 may lead to undesired results, or require many iterations when having a poor head pose initialization. This would be the case in a low framerate tracking scenario.

Therefore, we will propose an alternative in the following sections.

3.4.1 Proposed algorithm

To address the aforementioned points, an online framework is here proposed. The principle is to simply track the head pose using the algorithm described in Section 3.3, relying on the current best estimate of the person specific face model (α coefficients). At the moment of initialization, the best guess is $\alpha = \mathbf{0}$, i.e., the mean face. Then, whenever new data, which could help refine the subject specific face model, is available, the non rigid ICP 3DMM fitting (Section 3.2) is used to obtain a new estimate on the α parameters, thus improving the subject's face model and, as a consequence, improving the head pose tracking accuracy.

The proposed methodology is summarized in Algorithm 3. In the following, we will explain the relevant points.

Algorithm 3 : Online head pose tracking and 3DMM fitting.

1: Initialize $\alpha \leftarrow \mathbf{0}$ and $\Gamma \leftarrow \{\}$.
2: **while** data streaming **do**
3: Get input data $\mathbf{U}_t$
4: Retrieve face model $\mathbf{V}(\alpha)$
5: Estimate head pose $\{\mathbf{R}_t, \mathbf{t}_t\}$ based on $\{\mathbf{R}_{t-1}, \mathbf{t}_{t-1}\}$ and $\mathbf{V}(\alpha)$ (Sec.3.3). ▷ Head pose
6: **if** $\{\mathbf{R}_t, \mathbf{t}_t, \mathbf{U}_t\}$ is relevant to Γ **then**
7: Add sample $\{\mathbf{R}_t, \mathbf{t}_t, \mathbf{U}_t\}$ to Γ
8: Define Λ (stiffness set), based on Γ
9: Refine $\{\alpha, \Gamma\}$ using Algorithm 2, with parameters Λ. ▷ 3DMM fitting
10: **end if**
11: **end while**

Samples set Γ and selection criteria

As motivated in Section 3.2, the non rigid ICP 3DMM fitting profit from multiple observations, as they jointly support the estimation of the face model under noisy and incomplete depth data. Therefore, we rely on past data samples to fit the 3DMM in a batch processing, along with the new sample. Nevertheless, the computational cost of evaluating the change of parameters within the 3DMM fitting optimization grows linearly with the amount of data samples, meaning it is not possible to use all observed data up to time t. In addition, this is not necessary, as justified previously. We instead maintain a limited set of samples Γ.

The set Γ thus contains data observations $\{\mathbf{U}\}$, along with their estimated head pose. In this work, we used the head pitch and yaw angles to characterize the samples in Γ and to decide on whether a new observation should be included in Γ.

Therefore, a priori, we define a fixed set of *desired* head poses. If a new sample's head pitch and yaw angles are at a distance to one of the desired head poses which is lower than a threshold, the input data is considered as relevant for the 3DMM fitting (line 6, Algorithm 3). Note that these values are provided by the result of the head pose tracking step. If the data is indeed relevant, it is added to the set Γ and the 3DMM fitting algorithm is used. Notice that the algorithm not only re-estimates α, but it also refines their estimates on the head pose, based on the ICP cost. For efficiency, when fitting the 3DMM, the algorithm is initialized from the current best estimate of the parameters.

In future work, the selection criteria could be extended to discriminate according to the face expression (neutral expression is desirable), head motion, and the distance to the sensor, as the depth data accuracy increases as the person is closer to the sensor.

Stiffness set Λ

The stiffness set Λ (see Algorithm 2, line 3) is redefined before employing the 3DMM fitting algorithm (line 8, Algorithm 3). Recall that, the values in Λ define the amount of regularization

on the model deformations, as the stiffness parameter foster the values of α to be close to $\mathbf{0}$. Therefore, in our implementation, we defined a rule based methodology to set the values of Λ according to the samples included in Γ. We considered the following elements:

1. **Small set** Γ. For a small set of samples in Γ, Λ is assigned with large values. Therefore, limiting the amount of deformations.
2. **Yaw diversity in** Γ. We reduce the values in Γ if there is diversity on the yaw angles, as, in practice, self occlusion is more common along this dimension.
3. **Larget set** Γ. For a large set of samples, the values in Γ are the smallest. Thus, allowing the model to match closely the data.

At each call of the 3DMM fitting algorithm, Γ normally contains 1 or 2 values.

3.4.2 Point-to-plane 3DMM fitting

We here propose to use point-to-plane constraints for the 3DMM fitting. Empirically, we found the point-to-point constraints of the data term (cf. Equation 3.12) may lead the algorithm to converge to unsatisfactory results. Bad correspondences and poor initialization may have an impact on this, whereas a point-to-plane cost is normally more robust to these elements [Rusinkiewicz and Levoy, 2001].

To modify the 3DMM fitting to consider a point-to-plane cost, it is only necessary to redefine the distance computation of the data term (in contrast to Equation 3.12) as follows:

$$F_{d_i}^{\ j}(\mathbf{X}) := w_i^{0.5}\mathbf{n}_i(\alpha,\mathbf{R}_j,\mathbf{t}_j)^\top\left(\mathbf{R}_j(\mu_i+\mathbf{M}_i\alpha)+\mathbf{t}_j-\mathbf{u}_{C_j(i)}\right), \tag{3.23}$$

where $\mathbf{n}_i(\alpha,\mathbf{R}_j,\mathbf{t}_j)$ is the normal for point i in the mesh $T(\mathbf{V}(\alpha);\mathbf{R}_j,\mathbf{t}_j)$. In practice, we approximate this term. Its value is computed in step 7 of Algorithm 2, but it is kept fixed during the Gauss-Newton optimization (step 11, Algorithm 2).

The Jacobian needs also to be modified to consider the point-to-plane cost. We can define the point-to-plane Jacobian as a modification of the Jacobian computed using point-to-point constraints, shown in Equation 3.16. Let $\mathbf{N}^j$ be a block diagonal matrix of size $N_v \times 3N_v$. This matrix contains the normals of $T(\mathbf{V}(\alpha);\mathbf{R}_j,\mathbf{t}_j)$, assigning one normal per row and each normal occupying 3 columns (as $\mathbf{n}=(\mathbf{n}_x,\mathbf{n}_y,\mathbf{n}_z)^\top$). Therefore, the modification applies only to the data term (F_d) related entries (see Equations 3.15 and 3.16) as follows:

$$F_{d_{plane}}^{\ j} = \mathbf{N}^j F_{d_{point}}^{\ j} \qquad and \qquad \frac{\partial F_{d_{plane}}^{\ j}}{\partial \mathbf{x}} = \mathbf{N}^j \frac{\partial F_{d_{point}}^{\ j}}{\partial \mathbf{x}} \tag{3.24}$$

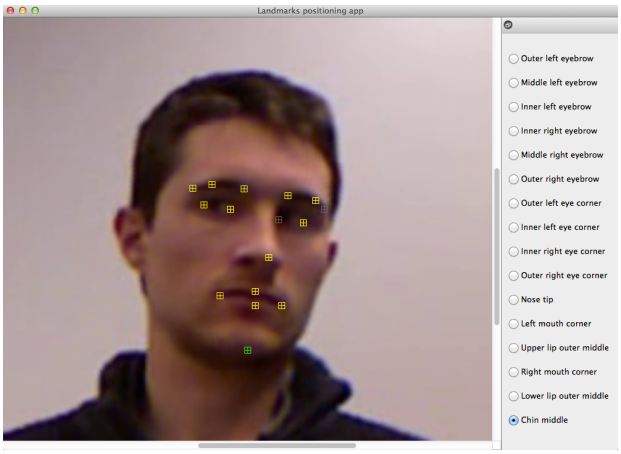


Figure 3.2: Facial landmarks manual annotations example

where "**x**" is a placeholder for either of the optimized parameters.

The landmarks term is maintained with point-to-point constraints, as these represent specific semantic correspondences, defined as 3D points.

3.4.3 Implementation considerations

The ultimate goal of this algorithm is the head pose tracking. To consider the possibly slow convergence of the batch 3DMM fitting, its execution runs in a separate thread from the head pose tracking. Nevertheless, it delivers the updates of the α parameters (updated face model), as soon as it is available to head pose tracker. Similarly, the head pose tracker provides new observations to the 3DMM fitting thread as soon as they are processed.

These considerations allow for the head pose tracker to run without interruption and to quickly improve the accuracy after initialization.

3.5 Experiments

In this section we present the experiments we conducted to validate the proposed methodologies. We will first provide information on the system that was implemented. Then, we will present experiments conducted on two publicly available datasets.

3.5.1 Implementation details and speed

The 3DMM we used is the Basel Face Model (BFM) [Paysan et al., 2009]. This model contains 53490 vertices and has 200 deformation modes. The BFM was learned from high resolution 3D scans of 200 individuals (composed of 100 male and 100 female participants) thus it spans a large variety of face shapes with neutral expression. We did not have a 3DMM for facial

expressions, thus we did not evaluate this aspect of the approach.

Offline non rigid 3DMM fitting

For the offline non rigid 3DMM fitting algorithm (cf. Section 3.2), we used the BFM's first 100 modes. The majority of the face region is used for the fitting, but we ignored the ears and neck regions, resulting in a mesh with 41585 vertices. The γ parameter was set as $0.5\frac{N_v}{Card(L)}$, such that the landmarks term weight has 0.5 times the cost of the data term, taking into account the number of data points-landmarks ratio. The λ_0 value (cf. line 3, Algorithm 2) was set empirically, such that its initial value is high enough to keep the α parameters close to $\mathbf{0}$ ($\lambda_0 = 0.1$ in our implementation) then $\lambda_{n+1} = 0.5\lambda_n$ within the iterative process.

Given a few annotated frames with landmarks as shown in Fig. 3.2 (typically 1 to 5 frames), the fitting algorithm takes from 5 to 20 seconds to optimize. Note that since people face shape is not expected to change much, this step is only performed once per subject, which means that the fitted model can be reused across sessions.

Head pose tracking

From the 3DMM instance, used as person specific face model, we used only 1710 points from the upper face region, defined a priori within the BFM topology (as shown in Fig. 3.1). Our CPU-based implementation[4] runs at $\approx$ 12fps. Nevertheless, the speed may decrease under large head movements which may require more ICP iterations. A careful GPU-based implementation could greatly increase the speed, as many of the computations could be well conducted in parallel, e.g., the correspondence search.

Online head pose tracking and 3DMM fitting

Within the online implementation, the head pose tracker itself remains unchanged, as the 3DMM fitting was implemented to run in a different thread.

Regarding the 3DMM fitting algorithm, it was reimplemented to take into account the different modifications, in particular the addition of the point-to-plane cost and the reduction of the overhead each time the 3DMM fitting is called by initializing from the previous estimates. The amount of 3D vertices used for the fitting was reduced to $\approx$15% of the points used in the offline case, to speed up the process. Therefore, for a single sample in Γ, it takes around 1-2 seconds to fit, whereas it takes $\approx$15-30 seconds for 9 samples.

The set of samples Γ, used for the 3DMM fitting, was limited to 9 samples maximum. The desired head poses to collect were defined such that Γ contains variations in terms of pitch and yaw angles around the frontal pose.

[4] The system was mainly implemented using Python and C++

The landmarks term was not exploited, as there are no landmarks estimates in our implementation yet. In some situations, this leads to person specific face models which are not semantically accurate, but are still a tight fit to the subject's face, and thus are sufficient for the head pose tracking.

3.5.2 Experimental protocol

To evaluate the proposed head pose tracking methods, we conducted experiments on two publicly available benchmarks, namely the BIWI kinect head database [Fanelli et al., 2011] and the ICT 3D head pose dataset (ICT-3DHP) [Baltrusaitis et al., 2012]. Both datasets were recorded with a Microsoft Kinect at VGA resolution (RGB and Depth). The BIWI dataset was annotated using the faceshift software[5], whereas the ICT-3DHP dataset uses the Polhemus Fastrack flock of birds tracker for its ground truth.

We therefore compare the two main strategies: the head pose tracker relying on a person specific face model built *offline* (cf. Sections 3.2 and 3.3), and the head pose tracker based on the *online* construction of the person specific face model, described in Section 3.4.

To evaluate if there is indeed a benefit on using a face model specific to each user, we also compared our results to the case in which the head pose tracker (cf. Section 3.3) use as face model the average face shape of the Basel Face Model (i.e., assuming $\alpha = \mathbf{0}$).

The offline approach was evaluated as follows: 3 to 5 RGB-D samples were manually collected, based on the variation of the head yaw and making sure the participant had a neutral face expression. These samples were then annotated with facial landmarks and processed to build the person specific face model from the 3DMM fitting algorithm. The head pose tracker was then used in the corresponding session and the results were compared to the ground truth.

The online approach was evaluated as follows: for a given subject/session we first run the online method to generate the person specific face model without any supervision. Then, the head pose tracker was run again in the corresponding session, from which the head poses estimates were extracted and compared to the ground truth. Notice that this is a valid approach, as we aim to evaluate the accuracy the method may achieve after the 3DMM fitting transcient period, and also the sequences in these datasets are short.

3.5.3 Results

The results obtained for the head pose tracking experiments are reported in Tables 3.1 and 3.2. In addition, we show the results from two alternative methods, namely Regression Forest [Fanelli et al., 2011], and CLM-Z with GAVAM [Baltrusaitis et al., 2012] which is a fitting method relying on both depth and RGB data. The performance reported for these methods was obtained from the experiments conducted by Baltrusaitis et al. [2012].

[5] www.faceshift.com

Table 3.1: Head pose tracking mean absolute angular errors obtained for the BIWI dataset. The Regression Forest method is from [Fanelli et al., 2011] and the CLM-Z with GAVAM from [Baltrusaitis et al., 2012]. "24/24" indicates that all sessions were used in the comparison, whereas "20/24" denotes 4 sessions out of 24 were discarded. Notice that the same sessions are used across methods, such that the reported results are comparable.

Method	Sessions	Yaw	Pitch	Roll	Mean
Regression forests	24/24	9.2	8.5	8.0	8.6
CLM-Z with GAVAM	24/24	6.29	5.10	11.29	7.56
Proposed (mean face model)	24/24	4.53	2.76	3.95	3.75
Proposed (offline face model fitting)	24/24	2.43	1.91	2.67	2.34
Proposed (online face model fitting)	24/24	4.94	2.60	3.58	3.71
Proposed (mean face model)	20/24	2.54	2.39	2.90	2.61
Proposed (offline face model fitting)	20/24	1.55	1.87	2.11	1.84
Proposed (online face model fitting)	20/24	1.41	1.58	1.76	1.59

Table 3.2: Head pose tracking mean absolute angular errors obtained for the ICT-3DHP dataset

Method	Sessions	Yaw	Pitch	Roll	Mean
Regression forests	10/10	7.12	9.40	7.53	8.03
CLM-Z with GAVAM	10/10	2.9	3.14	3.17	3.07
Proposed (mean face model)	10/10	4.44	2.78	4.13	3.78
Proposed (offline face model fitting)	10/10	3.61	2.25	3.61	3.16
Proposed (online face model fitting)	10/10	2.67	2.05	3.27	2.66
Proposed (mean face model)	8/10	2.61	2.20	3.60	2.80
Proposed (offline face model fitting)	8/10	2.39	2.08	3.45	2.64
Proposed (online face model fitting)	8/10	2.54	1.97	3.40	2.64

We can observe that our head pose tracking method has by far the lowest error for the BIWI dataset. Nevertheless, as discussed in Section 3.3.3, the failure detection method we used is prone to errors. There are a few sessions in which the head pose tracking diverged largely from the ground truth head pose, and remained at the wrong position for a period of time. In particular, for some sessions of the BIWI database, we encountered extreme head poses for which there were no depth measurements in the upper face region and caused the tracker to get lost (recall our method tracks only this region). In these situations, the reported average error does not accurately reflect the method's performance (in terms of pose accuracy, rather than tracking performance). Therefore, in Tables 3.1 and 3.2 we show the results obtained for the full database and also when discarding a *few* sessions, where this situation had an important impact on the results.

In general, the online methodology obtains higher or similar accuracy than the offline case, and, as expected, using the mean face model leads to the worse results of the three. If we ignore 4 sessions (out of 24) of the BIWI database, the mean angular error reduces to 1.84° for the offline case and 1.59° for the online case. However, note that the annotations for this

dataset were obtained using a head pose tracking method similar to ours (faceshift, Weise et al. [2011]), applied to the full face region. Therefore, with this evaluation, we can only conclude that our tracker obtains comparable results to this software.

For the ICT-3DHP dataset, our tracker achieves comparable results to the CLM-Z with GAVAM method for the offline model and slighly better results for the online method. However, again our approach had difficulties to recover from a faulty estimation in 2 out of the 10 sessions. In one of them, the subject's hair caused important errors. This suggest that a better outlier detection strategy would be beneficial in future work. If these sessions are ignored, both online and offline approaches achieve 2.64° of head pose estimation error.

It is important to mention that the evaluation of our tracker is slighly affected due to ground truth mis-synchronization. More precisely, while the ground truth has been synchronized with the RGB video, the RGB video happens to lose synchrony with depth in some sequences. This is a problem, as our tracker is purely based on depth (whereas the CLM-Z GAVAM method relies as well on the RGB data), causing misleading errors, in particular, during fast head movements. Nevertheless, the results obtained on both the BIWI and ICT-3DHP datasets demonstrate the potential of our head pose tracker in terms of head pose estimation accuracy.

Finally, note that accurate head pose estimation is crucial for further processing, as it will potentially impact the gaze estimation step in two ways. First, as a direct input to the estimation of the line of sight (LoS) in the 3D space. In this case, an error made in the head pose estimation almost immediately adds as an error to the computed gaze estimation. Second, in the extraction of the eye images based on the estimated head pose. Under head pose errors, this might lead to a frame by frame inconsistent eye image extraction. Eventually, this will introduce noise in the input data to the gaze tracking approach.

More details on these elements will be described in the following chapters. Nevertheless, to qualitatively illustrate the potential impact of head pose tracking on gaze estimation (in particular, for the second point), we present in Fig. 3.3, for a representative sequence, the eye cropping resulting from using the offline 3DMM fitted approach or using the mean face shape. As can be seen, since the mean shape does not fit well the given subject (although the results are still good), the pose tracking results oscillate even for similar head poses, generating an inconsistent frame by frame cropping of the eye image. Notice in contrast that more stable results were obtained when using a personalized face model. We observed a similar behavior when using the online face model fitting: the retrieved eyes images position are stable during tracking, although, in some cases, the eyes locations are not well centered. Nevertheless, overall, the better tracking results validate the use of a personalized template over the simple use of the mean face shape.

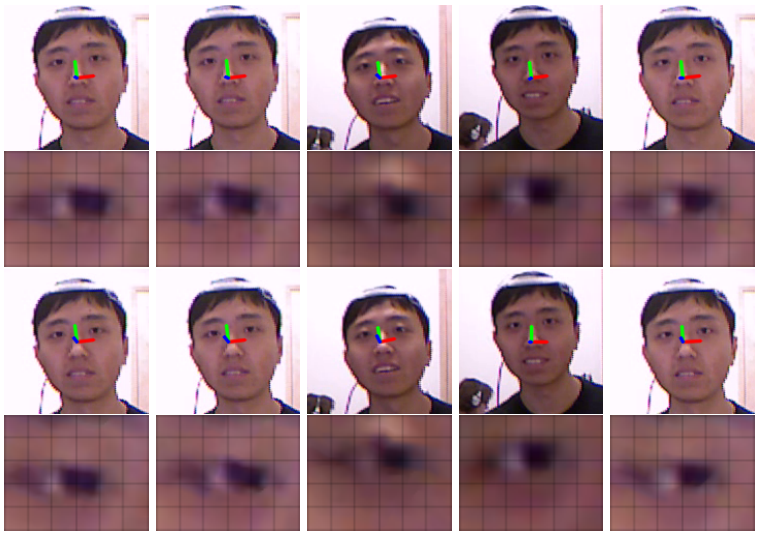


Figure 3.3: Impact of using a personalized face model on the eye image cropping. In this image, the coloured coordinate system is at the tip of the nose of the face model, and oriented according to the estimated head pose. The eye images are cropped and processed using the methodology described in Chapter 5. Each column depicts a different frame from a sequence from the ICT-3DHP database. The first 2 rows correspond to the results obtained when using a personalized face model fitted offline, while the two last rows show the results using only the mean face shape. As can be seen, in this later case, the rectification exhibit more inconsistent eye cropping in both the vertical and horizontal directions, which would negatively impact the gaze estimation process.

3.6 Discussion and future work

In this Chapter we have presented a head pose tracking methodology based on RGB-D sensors. The main principle is based on the iterative closest points (ICP) algorithm to continuously register a person specific face model to depth data.

We therefore presented two strategies to create the person specific face models. Both of them are based on a non rigid formulation of ICP, developed to fit a 3DMM to multiple snapshots of the subject. The advantage of fitting the 3DMM jointly to multiple data samples is that it allows to compensate for the device depth noise and missing depth data.

The first strategy is an offline approach, which requires supervision from the participant or from an experimenter (a third person in need to process the videos) to collect sample frames and to annotate facial landmarks. The 3DMM is then fit to those samples and the resulting face model can be used for the head pose tracking task.

The second strategy is based on an online formulation. The method jointly fits the 3DMM and conducts the head pose tracking task. Based on the head pose estimates, we collect samples adequate for the 3DMM fitting. In parallel, the more samples are observed and fitted with the 3DMM, the better the person specific face model is, thus improving the accuracy of the head

pose tracking. This formulation is more adequate for non cooperative and online scenarios.

The methods have been evaluated on two publicly available benchmarks. The results demonstrated the system is highly accurate. It is indeed accurate enough to be used in the gaze tracking task, which will be discussed in the following chapters.

For future work, there are many elements which can be addressed to improve this system, in terms of accuracy and practical limitations. First, the offline case required manual landmarks annotations, whereas in the online formulation, the landmarks were discarded in our experiments. An improvement may consider to integrate an automatic landmark detection method. In recent years there have been significant advances in this problem, which have potential to cope with low resolution and challenging head poses [Baltrusaitis et al., 2012, Dantone et al., 2012, Zhu and Ramanan, 2012, Xiong and De la Torre Frade, 2013, Cao et al., 2013, Kazemi and Sullivan, 2014].

Due to lack of a 3DMM with facial expressions, we did not evaluate facial expressions handling. Nevertheless, the system can be well extended to this situation, as explained in Section 3.2.3. Alternatively, in the online scenario we can use the automatic facial landmarks detector as a facial expressions proxy, from which the samples selection may be constrained to those with a neutral facial expression.

The visual domain can also help to improve the accuracy, as done by Morency and Darrell [2002], by integrating normal flow constraints. The normal flow constraints may benefit from the more accurate face model of the subject. In addition, skin color models segmentation may help to filter out bad correspondences. In particular, in the case of hair occluding the face, which can not be well discarded by the robust estimators.

4 EYEDIAP database

In this Chapter we describe in detail the EYEDIAP database, which we have collected and made available to the research community[1]. This work was published in [Funes Mora et al., 2014a], together with an accompaning technical report [Funes Mora et al., 2014b].

The objective of this Chapter is not only to provide a description of the EYEDIAP database itself, but also to define a common notation for researchers, in particular the users of this database, to characterize their experiments. The ultimate goal is to provide the community a framework which allows a direct objective comparison between gaze estimation algorithms.

4.1 Motivation

In the past years many methods have been proposed for the task of gaze estimation. We have presented in Chapter 2, Section 2.2, a literature review covering the wide diversity of research that has been conducted in this field. Even though the evaluation methodologies employed by researchers have clearly advanced the development of gaze tracking algorithms, and have well validated their contributions, it is unlikely to encounter in the literature comparisons evaluated on the same data. This is particularly true for gaze estimation methods relying on natural light based sensing, be it a geometric based or appearance based approach.

This makes it difficult to clearly identify the advantages and disadvantages of each one of the proposed methods. The main reason is the lack of standard benchmarks under which researchers can train and/or evaluate their algorithms, and report their results.

Therefore, to address the need of the research community, and to develop an adequate framework for us to investigate the different aspects of the gaze estimation problem, we created the *EYEDIAP* database. The availability of this database has been crucial for the development and validation of the contributions of this thesis.

This database was collected with consumer RGB-D sensors, together with a high resolution

[1] The EYEDIAP database may be downloaded from the following url: https://www.idiap.ch/dataset/eyediap

camera. It was designed to be representative of a wide spectrum of scenarios and sensing conditions, such as the ones depicted in Figure 1.1. The recording methodology systematically includes, and isolate, most of the variables which affect remote sensing based gaze estimation algorithms, such as head pose, participant, ambient conditions and the participant's gaze behavior. This allows to define benchmarks over which to investigate on each of these elements separately. This will be described in detail in the following sections.

Notice that recent efforts were made by other researchers to create gaze estimation datasets. The Columbia gaze dataset is one example, which was created by Smith et al. [2013]. This dataset is a promising resource to advance the research on gaze estimation, in particular, for the training and evaluation of appearance based gaze estimation methods, as first used by Schneider et al. [2014]. It was carefully collected and contains a large quantity of participants (56). However, it has several limitations: the range of head poses and gaze directions is small, there is no temporal information (only static images), they only provide RGB images, and, as gaze targets, it is limited to only 21 points defined on a plane at a fixed distance. More recently, Sugano et al. [2014] released a similar database to Smith et al. [2013]. Their database is more dense, in terms of head poses and gaze directions. Furthermore, during the collection, they used a multi-camera setup to then synthesize additional viewpoints using 3D reconstruction, resulting in a much larger database. However, the range of gaze directions is still small, in great part because this database is restricted to gaze targets defined as points within a computer screen and, similar to Smith et al. [2013], only static images are provided.

Even though these datasets are indeed valuable resources to the research community, the EYEDIAP database does not have many of their aforementioned limitations. Our database was designed to be flexible, in terms of its usage, and it is representative of diverse sensing conditions and scenarios.

The rest of the Chapter is structured as follows: in Section 4.2 we present the database design and its collection procedure. Section 4.3 describes the methodology we employed to prepare the provided metadata data. In Section 4.4 we define the general evaluation protocol and measures. In Section 4.5 we propose different benchmarks, over which it is possible to evaluate and characterize gaze estimation algorithms. Finally, in Section 4.6 we conclude the chapter.

4.2 Data collection and design

In this section we first describe the recording methodology. We will then describe the recording sessions that were designed to constitute the database.

4.2.1 Setup

The recording setup is as shown in Fig. 4.1. It comprises an RGB-D camera (a Microsoft Kinect™), an HD camera, an ensemble of 5 LEDs located within the field of view of both

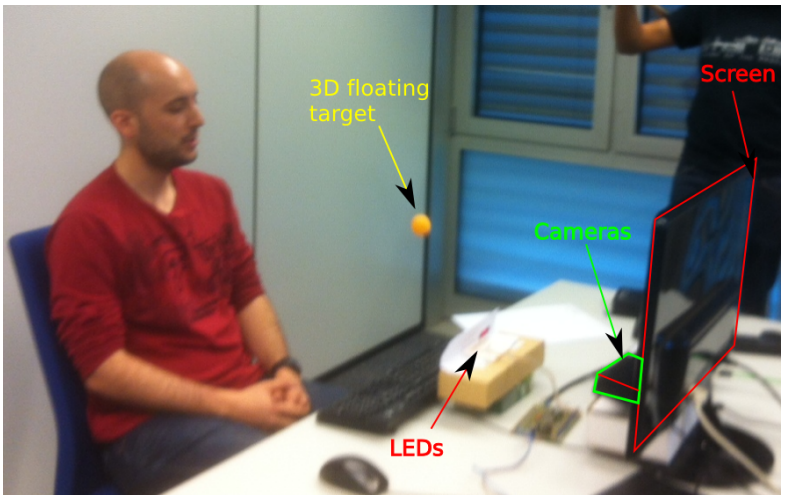


Figure 4.1: EYEDIAP database recording setup

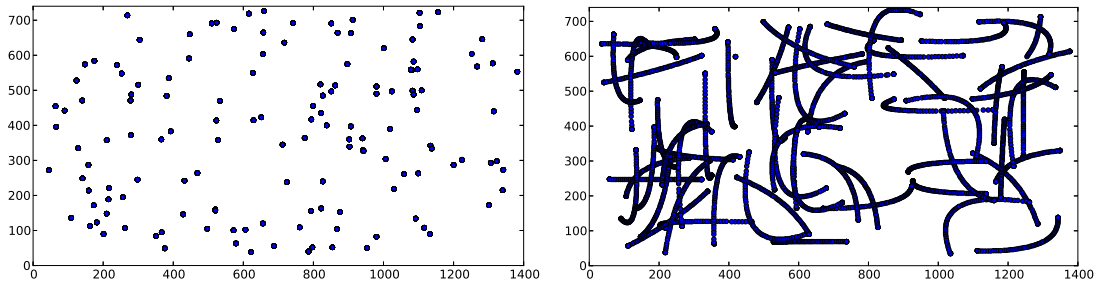
cameras, a 24" flat computer screen and a 4cm diameter ball, which was used as a visual target for some of the recordings. The characteristics and purpose or function of each element are described as follows:

- **Microsoft Kinect for Xbox**: this consumer RGB-D sensor provides standard video (RGB) and depth video (D) streams, both at VGA resolution (640 × 480) and at a 30 frames-per-second acquisition rate. The data was acquired using the *libfreenect* library [2].
- **HD camera**: the Kinect was designed with a large field of view for full body capture, which imposes less restriction on user mobility but is problematic for eye tracking based on VGA resolution, as it leads to low resolution eye images. Therefore, we also recorded the scene with a full resolution HD camera (1920x1080) at 25fps. This camera was positioned as close as possible to the Kinect sensor to capture a similar viewpoint.
- **LEDs**: we placed 5 LEDs which were visible by both cameras. Their purpose is to help synchronize the RGB-D and HD video streams using a time changing binary code displayed by the LEDs (see Section 4.3.4).
- **Flat screen**: we used a 24" computer screen to display a visual target (see Section 4.2.2). The effective screen resolution, i.e., the region used to display the visual target, was of 1340 × 740.
- **Small ball**: we used a 4cm diameter ball as a visual target with a double purpose: to serve as a visual target in a 3D environment and, to be discriminative in both RGB and depth data, such that its 3D position could be precisely and reliably tracked (see Section 4.3.6).

As shown in Fig. 4.1, the cameras are right below the computer screen. This is intended to observe the participant's eyes from below, thus minimizing eyelids occlusions.

We now describe the designed recording sessions that were recorded using this setup.

[2] http://github.com/OpenKinect/libfreenect



(a) (b)

Figure 4.2: Example of screen visual target coordinates during a recording session using either the a) discrete screen target (DS); or b) continuous screen target (CS).

4.2.2 Recording session

For each recording, the participant was requested to sit in front of the setup, within the field of view of the cameras. Instructions were then given to gaze at the specified *visual target* throughout the duration of the recording session. No restrictions were made in terms of speaking activity, facial expressions, blinks, etc. This, in order to encourage natural behavior.

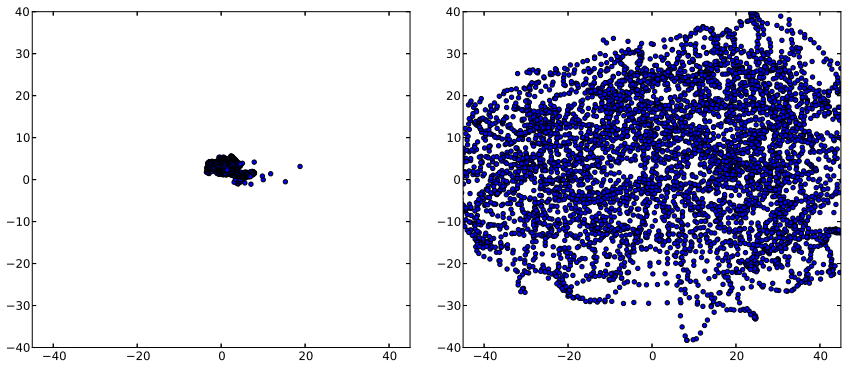
In order to allow the systematic evaluation of gaze estimation algorithms, we designed the database such that each one of the recording sessions may be characterized by a combination of the main variables that affect gaze estimation accuracy.

The four variables we considered are the type of visual target, the head pose activity, the participant and the ambient conditions. We describe how we addressed each one of these variants as follows.

Visual target

The visual target is the object the participant was requested to gaze at during the recording session. In order for this database to be representative of diverse applications, we included the following types of visual target:

- **Discrete screen target (*DS*)**. In this case, a small circle was drawn at random locations in the computer screen. The screen coordinates were sampled from a uniform distribution, and changed every 1.1 seconds. See Fig. 4.2a for an illustration. This target would encourage saccadic eye movements, and fixations to the target position during a short period of time.
- **Continuous screen target (*CS*)**. Similarly to the DS case, a small circle was drawn in the computer screen but programmed to move along a trajectory parameterized by a quadratic Bézier curve. The control points of this curve were drawn randomly from a uniform distribution, defined within a smaller region of the screen. A new trajectory was redefined every 2 seconds. See Fig. 4.2b for an example. This target was intended to encourage a smooth



(a) (b)

Figure 4.3: Frame by frame head pose observations for one session (15-B-FT-*H*), where *H* corresponds to the head pose activity, either (a) *Static* or (b) *Moving*. In both cases, the euler angles of the head pose are shown: yaw in the *x* axis and pitch in the *y* axis.

pursuit behavior while the participant gaze at the moving target, in addition to the saccadic eye movement whenever a new path is defined.

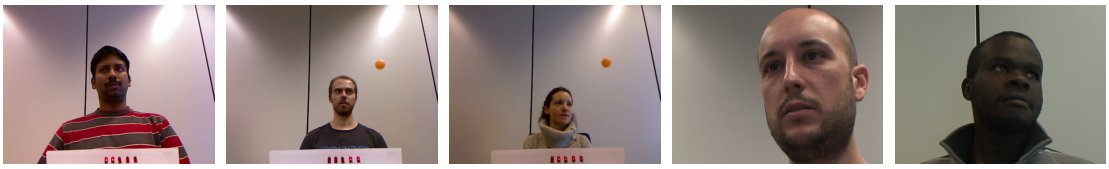
- **3D floating target (*FT*)**. This corresponds to a 4cm diameter ball hanging from a thin thread attached to a stick, and that was moved within a 3D region between the camera and the participant. In contrast to the screen based targets, the participant was at a larger distance to the camera to allow for sufficient space for the target mobility. This object is discriminative in both color and depth, allowing to retrieve its 3D position automatically (see Section 4.3). The FT visual target mostly leads to smooth pursuit gaze behavior. Nevertheless, the target movements lead to a much larger coverage of the 3D space. This encourages a much larger range of gaze directions and, possibly, of head pose variations. Furthermore, this target is representative of much less constrained scenarios defined in the 3D space (see Fig. 1.1).

Notice that the visual target type imposes spatial constraints on the setup. For the 3D floating target case, the distance to the recording sensor was around 1.2m, as this was necessary to allow the ball to move in front of the participant, and to be sensed by the Kinect. For the screen based targets, the participant could be closer, at a distance of approximately 80-90cm from the recording sensor. At closer distances the Kinect would not provide depth measurements.

Head pose activity

In order to evaluate methods in terms of robustness to head pose variations, we requested the participant to keep gazing at the visual target while performing one of the two following head pose activities:

- **Static (*S*)**. In this case, the participant was asked to keep an approximately static head pose, facing towards the screen (and cameras) during the recording. Fig. 4.3a shows an example



(a) (b) (c) (d) (e)

Figure 4.4: Examples of the recorded data using: (a)-(c) the RGB-D camera; and (d)-(e)) the HD camera, for which the images were cropped to a size of 640 × 480 for display comparison with the VGA resolution data. In these examples the participant is: (a),(d) gazing at the screen target with a static head pose; (b) gazing at the floating target with a static head pose; (c),(e) gazing at the floating target while moving the head.

of the obtained distribution of head pitch and yaw angles for one recording session.

- **Moving (*M*)**. For this condition, the participants were asked to perform head movements in order to introduce head pose variations. We encouraged variations in terms of both rotations and translations. However, in practice, the majority of head movements were due to rotations. In Fig. 4.3b we show an example of the empirical distribution of head pitch and yaw angles for one recording session. As it can be observed, the obtained head poses are rich in terms of variations.

Participants

Our dataset was recorded for 16 people: 12 males and 4 females, with age ranging from 20 and 45 years. The participants are from diverse origin, with a total of 12 nationalities, e.g., 2 people from Central America, 1 black African, 4 caucasian French, 1 Indian, 2 caucasian Swiss, etc. As a result, the eye shapes and appearance exhibit a large variability. Each participant was assigned an identifier, from "1" to "16".

Ambient conditions

For participant 12, 13 and 14, the sessions involving the *FT* target were recorded twice. Nevertheless, the ambient conditions were significantly different between these recording sessions: different day, illumination and distance to the camera. To distinguish among these situations, we denote the two possible conditions as "*A*" or "*B*".

4.2.3 Summary of the collected data

In total we recorded 94 sessions. Each session may be characterized by the string "P-C-T-H" which refers to the participant identifier P=(1-16), the recording conditions C=(A or B), the employed visual target T=(DS, CS or FT) and the head pose activity H=(S or M). Examples of the recordings can be seen in Fig. 4.4.

Each session in conditions "*A*" correspond to 2.5 minutes of recording time, whereas the sessions recorded in conditions "*B*" last approximately 3 minutes each. This corresponds to more than 4 hours of data. We summarize all recorded data in Table 4.1. A complete list of the recording sessions is available at the database website[3].

Table 4.1: Summary of the recorded sessions.

Participants	Recorded sessions (the participant identifier is implicit)
1-11	A-DS-S; A-DS-M; A-CS-S; A-CS-M; A-FT-S; A-FT-M
12-13	B-FT-S; B-FT-M
14-16	A-DS-S; A-DS-M; A-CS-S; A-CS-M; A-FT-S; A-FT-M B-FT-S; B-FT-M

4.3 Data processing

Besides the collected data itself, described in Section 4.2, we provide metadata that is essential for deriving the gaze ground truth, and which is also useful to exploit the dataset, and to easily run experiments. It comprises the camera(s) calibration information, camera-screen calibration, the 3D head pose, the approximate 3D location of the eyes, the frame by frame 3D location of the visual target and manual annotations.

A visualization of the provided metadata is shown in Fig. 4.5. For the description on how to interpret the provided files, please refer to [Funes Mora et al., 2014b]. In the following, we describe the procedure we employed to estimate these parameters.

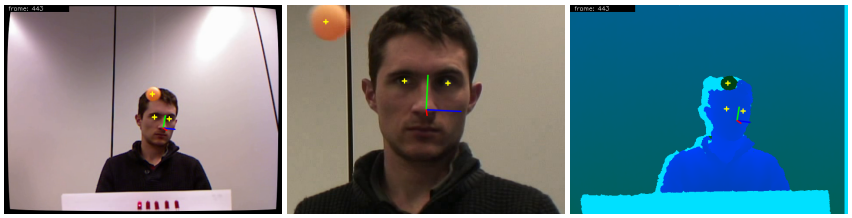
4.3.1 World coordinates system definition

To standardize the definition of all 3D variables in the data, we have defined a common world coordinate system (**WCS**). It was defined with a fix pose relative to the RGB camera of the Kinect, such that the participant is near the point $(0,0,0)^\top$ (roughly) and the axis are consistent with the OpenGL standard. If $\mathbf{p}_\kappa \in \mathbb{R}^3$ is a point defined with respect to the coordinate system of the Kinect RGB camera, then we have defined the **WCS** such that the equivalent point $\mathbf{p}_W$, with respect to the **WCS** is given by $\mathbf{p}_W = \mathbf{R}_W\mathbf{p}_\kappa + \mathbf{t}_W$ where[4]:

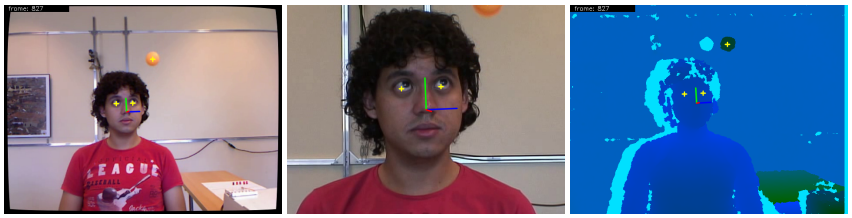
$$\mathbf{R}_W = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \mathbf{t}_W = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{4.1}$$

[3] https://www.idiap.ch/dataset/eyediap/sessions

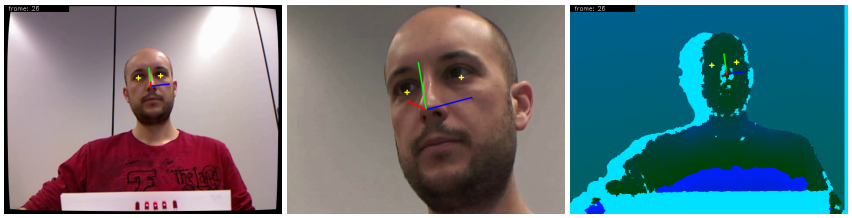
[4] The units are in meters.



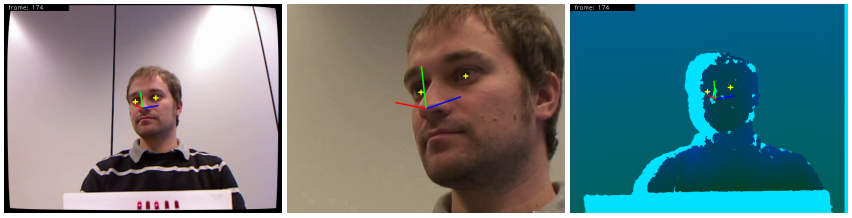
(a)



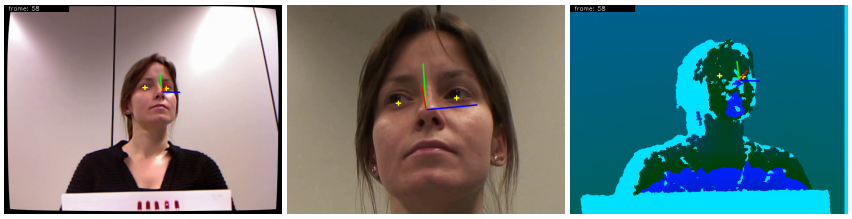
(b)



(c)



(d)



(e)

Figure 4.5: Examples of the processed data with head pose, eyes and floating target tracking. For each example we show, from left to right, the Kinect RGB frame, the HD frame (a 640 × 480 cropped region) and the depth frame encoded as an RGB image. The displayed head reference is positioned at a fixed distance from the **HCS** for visualization (not at the nose tip). The points drawn at the eyes correspond to the projection of the approximate 3D eyeball center into each respective image (cf. Section 4.3.5). When relevant, the found ball center is also shown projected into each image (cf. Section 4.3.6). (a) Frame 443 for session 1_A_FT_M; (b) Frame 827 for session 13_B_FT_S; (c) Frame 26 for session 3_A_DS_S; (d) Frame 174 for session 15_A_CS_M; and (e) Frame 58 for session 8_A_CS_M.

4.3.2 RGB-D sensor intrinsic calibration

The Kinect sensor was calibrated using the toolbox from Herrera C. et al. [2012]. This procedure generates the intrinsic parameters for both the depth and RGB cameras together with the extrinsic parameters, i.e., the 3D pose between both sensors, which in turn defines the depth sensor's pose with respect to our defined **WCS**. In addition, this toolbox estimates the mapping from depth disparity[5] to actual depth, including a correction for non linear distortions of the depth map, as proposed by Herrera C. et al. [2012].

The procedure to interpret the RGB-D data is described in detail in [Herrera C. et al., 2012], in particular the disparity to depth transformation. The RGB video we provide has been pre-processed to correct for non-linear distortions.

Notice that, according to Herrera C. et al. [2012], the typical depth error is around 3mm at a distance of $\approx 1.2m$, which is the case for the *FT* visual target sessions. For the screen based sessions (*CS* and *DS* targets), the typical depth calibration error would be around 1.5mm, corresponding to a distance of $\approx 0.85m$.

4.3.3 3D screen calibration

We calibrated the system to determine the pose of the screen with respect to the 3D world coordinate system (**WCS**), in addition to the transformation from screen pixel coordinates to meters. Therefore, provided the calibration parameters, we can formulate the mapping of a point $\mathbf{p} \in \mathbb{R}^3$, defined in the **WCS**, to the screen coordinates $\mathbf{s} \in \mathbb{R}^2$ as shown in Eq. 4.2.

$$\mathbf{s} = \begin{bmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \end{bmatrix} \left(\mathbf{R_s p} + \mathbf{t_s} \right), \tag{4.2}$$

where a 3D coordinate system $\mathbf{SCS} = \{\mathbf{R_s}, \mathbf{t_s}\}$ has been defined at the coordinates $(0,0)$ of the screen. The values k_x and k_y denote the pixels per *meter*[6] constants along the x and y coordinates respectively. Notice that Eq. 4.2 assumes that $\mathbf{p}$ already lays in the screen plane, by ignoring the z component once it is referred to **SCS**. The inverse transformation is straightforward: a 3D point $[\mathbf{s}^x / k_x, \mathbf{s}^y / k_y, 0]^\top$ is defined and transformed by $\mathbf{SCS}^{-1}$.

To infer the parameters of the transformation, i.e., $\mathbf{R_s}$, $\mathbf{t_s}$, k_x and k_y we designed a mirror-based technique: a mirror was located in front of the RGB-D camera, whose 3D plane was found by leveraging on markers placed on the mirror surface, which were observable in the depth map. The mirror was positioned such that the screen was visible by the RGB-D sensor. Furthermore, the depth map correctly measured the depth of the screen's reflection.

[5] Note the depth map provided by *libfreenect* is actually a disparity map, not meters or millimeters

[6] To meters, rather than millimeters, as mentioned in Funes Mora et al. [2014a]

We then displayed a colored discriminative target in the screen at coordinates $\mathbf{s}$. Notice that its reflection through the mirror was visible from the RGB-D sensor. The observed (virtual) target's position in the RGB image was found automatically. A color model is learned from manual annotations on a few frames. Its 2D position is then found through color likelihood evaluations in the RGB image, filtered using the corresponding depth measurements. Its 3D position could then be retrieved from depth data. Based on the mirror's plane and the law of reflection, this virtual 3D point is transformed back to its true position $\mathbf{p} \in \mathbb{R}^3$, defined with respect to the **WCS**. This process is repeated and used to collect a set of pairs $\{(\mathbf{s}, \mathbf{p})\}$. Then the transformation parameters are obtained as the least squares solution from an overdetermined linear system of equations based on Equation 4.2.

4.3.4 RGB-D and HD camera synchrony and calibration

In addition to the RGB camera from the Kinect, the session was recorded with an HD camera. Even though in this thesis we are interested in low resolution conditions, we still aimed to provide adequate data to develop and evaluate algorithms which rely on high resolution data.

Nevertheless, in order to use the HD data together with the provided metadata, and even with the depth video, it is necessary to have synchronization with the RGB-D video stream. Therefore, to achieve synchrony, we used a set of 5 LEDs which were activated in the order determined by the binary Gray Code, such that only one LED turns on or off at each transition. These LEDs were within the field of view of both cameras, and the goal was to post-process the data by aligning the code observed in both cameras. Retrieving the observed code was done using a Hidden Markov Model (HMM), with transition probabilities according to the Gray Code, and emission probabilities according to noisy observations of the LEDs color. Once aligned, the HD video was reencoded to be in full synchrony with the RGB-D stream.

Stereo calibration between the two cameras is also desired, as for example, to use the HD video with depth data. To this end we used the standard stereo calibration procedure using a chessboard pattern for cross features. The output from this method was the HD camera pose **DCS** $= \{\mathbf{R}_{HD}, \mathbf{t}_{HD}\}$, provided with respect to the **WCS**, and its intrinsic parameters, according to the pin-hole camera model. In this manner, a point $\mathbf{p}^H \in \mathbb{R}^3$ defined in the HD camera coordinate system is transformed to the **WCS** as $\mathbf{p}^W = \mathbf{R}_{HD}\mathbf{p}^H + \mathbf{t}_{HD}$.

4.3.5 Head pose and eyes tracking

For each participant we created a 3D person specific face model. This was done by fitting a 3D Morphable Model (3DMM) to RGB-D data, using the algorithm we described in Chapter 3, Section 3.2, based on the Basel Face Model [Paysan et al., 2009].

Provided the face model, we tracked the 3D head pose using the approach we described in Chapter 3, Section 3.3. The result is the estimated head pose for each frame, given as $\mathbf{p}_t = \{\mathbf{R}_{ht}, \mathbf{t}_{ht}\}$ of a 3D rotation and translation (with respect to the **WCS**).

From the 3DMM topology, an *approximate* location of the eyeball center can be defined a priori, and denoted as $\tilde{\mathbf{o}}$. To obtain $\tilde{\mathbf{o}}$, we used the average of the eyelids contour points, followed by a translation of 5mm towards the inside of the face model. Notice that $\tilde{\mathbf{o}}$ will be different for each person, as it depends on the 3DMM coefficients resulting from the fitting. Within the database we provide the head pose tracking results, and $\tilde{\mathbf{o}}$. In this manner, the 3D eyeball location at time t can be computed as:

$$\mathbf{o}_t = \mathbf{R}_{ht}\tilde{\mathbf{o}} + \mathbf{t}_{ht} \tag{4.3}$$

4.3.6 Floating target tracking

For the recording sessions using a ball as a visual target (*FT*), we provide the 3D center of the ball at every time step t, denoted as $\mathbf{b}_t \in \mathbb{R}^3$ and defined with respect to the **WCS**.

This value was computed as follows: the depth map was thresholded to remove the background; then, the 2D point with the maximum color likelihood is selected as the location of the ball target, where the color distribution of the target was learned from manually segmenting the ball target in one or two images. If the color likelihood, accumulated within a region defined a priori, according to the target's size, is smaller than a threshold, then the candidate is discarded. Otherwise, once found, a template 3D mesh, with the size and shape of the target, was rigidly registered to depth data using the iterative closest points algorithm. Finally, $\mathbf{b}_t$ is defined as the center of the registered template.

This approach ensures that the automatically retrieved ball target's 3D position is reliable and accurate. However, notice that the estimation of the target's position depends on the depth data. Thus, during the recording sessions, at moments when the target leaves the field of view of the camera or it gets too close to the depth sensor and reaches its sensing limit, the depth is not available for the target making it impossible to determine its 3D position.

4.3.7 Manual annotations

Further manual annotations were conducted. The purpose was to identity at which instants the automatically computed ground truth needs to be considered as unreliable, be it for the training or evaluation of gaze estimation methods.

Therefore, we annotated the moments during which there is an eye blink, or when the person is clearly distracted and not looking at the target. The manual annotations were done for all the sessions involving the *CS* and *FT* targets. We currently did not annotate the *DS* sessions. These further annotations could be made in the future, if needed.

4.4 Evaluation protocol and measures

In this section we discuss elements to consider when conducting experiments using the EYEDIAP database. We thus define the ground truth data and how it can be compared to the output of a gaze estimation algorithm. We also discuss which frames, within the session videos, are exploitable for the purpose of training and evaluation of gaze estimation methods. We then formally define the concept of a gaze estimation algorithm, followed by the notions of *train*, *test* and *evaluation* sets. Finally, we discuss the performance measures to be used.

4.4.1 Ground truth data and task

With respect to the quantity we define as ground truth $\hat{\mathbf{g}}$ for gaze estimation algorithms, this depends on the visual target that was used for a given session, as explained in the following.

Ground truth: 3D floating target based sessions (*FT*)

In this case $\hat{\mathbf{g}} := \mathbf{b}$ where $\mathbf{b}$ is the 3D position of the ball target, as computed in Section 4.3.6. Although $\mathbf{b}$ was obtained using an automatic method, its position was determined with high accuracy, as qualitatively shown in Figure 4.5. The use of both color and depth information makes the retrieved values reliable, with almost no false positives found in the database.

Ground truth: Screen based recording sessions (*CS* or *DS*)

In the case of screen based data, the 2D coordinates of the target displayed in the screen $\hat{\mathbf{s}} \in \mathbb{R}^2$ should be considered as the ground truth data. In other words, $\hat{\mathbf{g}} := \hat{\mathbf{s}}$, where it is assumed the participant is indeed fixating at the target $\hat{\mathbf{s}}$. As we described in Section 4.3.3, we can transform $\hat{\mathbf{s}}$ into a 3D point $\mathbf{p}$ with coordinates defined in the **WCS**. Although the screen-camera calibration parameters were obtained using a careful calibration procedure, it is not possible to determine its accuracy.

Notice that we do not consider the head pose tracking or related eyeball position as ground truth data. These are provided to facilitate the work of other researchers, which may profit from a prior estimation of these quantities. Nevertheless, this data should be understood as the result we obtained using our approach. Alternative methods could replace these elements.

Gaze estimation output and ground truth comparison

The output $\mathbf{g}$ of a gaze estimation method depends on its methodology, but it can be either the point of regard (*PoR*), as a 2D (screen coordinates) or a 3D point, or the 3D line of sight (*LoS*). When the output is the *LoS*, a gaze estimate includes the two following elements: the origin of the *LoS* gaze ray $\mathbf{o} \in \mathbb{R}^3$ (in principle, a point related to the eyeball) and the *unitary* vector $\mathbf{v} \in \mathbb{R}^3$, such that $\mathbf{g} = \{\mathbf{o}, \mathbf{v}\}$. Both quantities should be expressed in the **WCS**.

Note that, in the screen based recording sessions, it is possible to compute the screen pixel coordinates from the 3D *LoS*. To this end, the *LoS* intersection to the screen plane is estimated (a 3D point), which is transformed to screen coordinates using Eq. 4.2. Alternatively, from the *PoR*, in screen coordinates, it is possible to compute the equivalent *LoS*, assuming the origin point $\mathbf{o}$ is known. The gaze vector is computed as $\mathbf{v} \propto \mathbf{p} - \mathbf{o}$, where $\mathbf{p}$ is the equivalent 3D point in the screen. This is valuable to have a *LoS* representation for algorithms which compute directly the 2D *PoR*, which, as $\mathbf{o}$ value, may profit from the eyes tracking results we provide.

Therefore, in order to have a standard approach to evaluate the performance of a gaze estimation algorithm, regardless of its methodology or the used visual target, we propose to compare between the estimated and the ground truth lines-of-sight, regardless on whether they are computed directly or indirectly. Researchers may then rely on the provided metadata.

4.4.2 Valid frames

The EYEDIAP database is composed of non stop video recordings, in which not all frames are exploitable for the purpose of training and evaluation of gaze estimation algorithms. Therefore, when using the database for any of these purposes, it is important to remove frames for which, either the data or the ground truth is not reliable, due to any of the following reasons:

- **Unavailable ground truth**. Obviously, frames for which the ground truth is not available should be discarded. This includes the situations in which the *FT* target is outside the sensor's field of view, or when it is too close to the sensor, beyond its depth sensing limit.
- **Gaze shifts**. For the *DS* and *CS* sessions, the visual target periodically changes its position within the screen. Therefore, whenever a new position (*CS*) or a new trajectory (*DS*) is defined, it is recommended to ignore a few frames. In our experiments, we typically ignore *600-700ms* of data after a position jump to ensure that the participant is indeed gazing at the target. This is a conservative amount, but it is based on observing the data and measuring the reaction time of the participants, of which we ignore a few further frames to guarantee the participant is indeed fixating at the target in the valid frames.
- **Extreme head poses**. In the cases of extreme head poses, the visibility of the eyes may be heavily compromised, e.g., under extreme head yaw angles, one of the eyes will be heavily occluded by the nose. These samples can be automatically removed from the valid frames by thresholding the head pose euler angles. Notice that tests of this type may be applied separately to each eye.
- **Extreme gaze directions**. These are situations in which the head pose and the gaze target $\hat{\mathbf{g}}$ were well measured, but the resulting $\mathbf{v}^{\mathbf{g}t}$ (gaze ground truth direction) is almost anatomically impossible to achieve (e.g., gaze yaw beyond 45 degrees), making it unlikely that the person was actually gazing at the target.
- **Blinks and distractions**. These are samples which were manually labeled as outliers, because the person is blinking or clearly not gazing at the visual target.

4.4.3 Gaze estimation algorithm definition

A gaze estimation algorithm is denoted as a function $\mathscr{H}$ which, provided a training set $\mathbf{T}$ and test data $\mathbf{D} = \{\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_t, \ldots, \mathbf{I}_T\}$, outputs a set of gaze estimates $\mathbf{G} = \{\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_t, \ldots, \mathbf{g}_T\}$ with one to one correspondence to the elements in $\mathbf{D}$. This is shown in Eq. 4.4:

$$\mathbf{G} = \mathscr{H}(\mathbf{D}|\mathbf{T}) \tag{4.4}$$

The index t usually represents time, in which case, $\mathbf{D}$ contains a sequence of frames (video). This would be representative of algorithms relying on temporal information. Nevertheless, to consider methods which estimate gaze from single images as well, this is not required. Furthermore, $\mathscr{H}$ may or may not output the estimation of gaze for each eye separately.

In the following, we will define the different sets involved in the evaluation of $\mathscr{H}$.

4.4.4 Training set

The training set, formally defined as $\mathbf{T} := \{(\hat{\mathbf{I}}, \hat{\mathbf{g}})_{\hat{t}}\}_{\hat{t}=1}^{N}$, consist of a set of images[7] - gaze ground truth pairs. Here we consider different ways to collect these samples for a given experiment:

- **Temporal**. The training data $\mathbf{T}$ is assumed to be temporal, and often corresponds to a recording session or a temporal section of a recording session.
- **Structured**. The training data is collected in a structured manner in order to fulfill a specific requirement of the gaze estimation algorithm. This can be, for instance, to obtain a predefined number of points in a screen with desired $\hat{\mathbf{g}}$ values.

Notice that it is possible to build a structured training set by selecting specific samples from a temporal one, assuming there is sufficient data in the latter to fulfill the structured set constraints. For the rest of the discussion, it is therefore assumed a training set is always provided as temporal. It is up to the user to decide how to use the samples.

Furthermore, a larger training set can be created by joining training sets from different recording sessions, as needed.

4.4.5 Test set

The test data $\mathbf{D}$ here will be assumed to be, similarly to the training case, a temporal section of one or a set of recording sessions. Nevertheless, $\mathbf{D}$ *must* be a disjoint set from $\mathbf{T}$.

[7] "Image" here denotes any data modality (RGB, depth, HD) or combination of data modalities.

4.4.6 Evaluation set

Finally, given the test set, we further define the *evaluation* set **E**. This set comprises the frames of the test set on which the performance of an algorithm is computed.

The objective of defining **E**, is to remove samples from **D** when computing the performance measures. Such samples are removed because, for a given reason, would be known a priori that can not be well addressed by the given method, and therefore, discarded to avoid introducing unnecessary noise into the performance measures computation. We will here discuss a particular example, specific to methods which are based on interpolation techniques.

Convex hull based filtering

Many appearance based methods, due to their interpolation nature, are known a priori to be capable of estimating gaze only within the convex hull of the training set in terms of gaze directions. For this type of methods, the range of gaze directions should ideally be larger or equal for the training set than for the test set. However, depending on the experiment definition, such condition can not always be guaranteed.

Therefore, to allow evaluation and comparisons without being affected by obvious errors, to define **E**, the user may also discard the samples in **D** for which the gaze ground truth is outside the convex hull of the gaze directions of the training data. Even though in real life conditions the gaze estimation algorithm does not have access to the test data ground truth, we can assume that the system ultimately implementing such method, has a control on the methodology used to collect the training data, which should be designed ensuring that this condition is met.

Note that comparing the performance of methods considering or ignoring this point is interesting to distinguish algorithms which are capable of *gaze extrapolation* (typically, those that rely on a geometrical model) from those that can not.

4.4.7 Performance measures

In this section we define the main performance measures we used in this thesis to compare different algorithms. For an index t in the evaluation set **E**, with estimated gaze direction $(\mathbf{o}_t, \mathbf{v}_t)$, i.e., the *LoS*, computed independently of the gaze estimation methodology or the visual target type (see Section 4.4.1), we considered[8] the following error measures:

- **Angular error** ${\epsilon^\circ}_t$. The per-sample angular gaze estimation error is computed as:

$$ {\epsilon^\circ}_t = \arccos\left(\mathbf{v}_t \cdot \mathbf{v}_t^{\mathbf{g}t}\right), \tag{4.5} $$

  where $\mathbf{v}_t^{\mathbf{g}t}$ denotes the ground truth 3D gaze vector: a unitary vector pointing from $\mathbf{o}_t$ to

[8] Additional performance measures can be found in Appendix A.1

$\hat{\mathbf{p}}_t$, where $\hat{\mathbf{p}}_t \in \mathbb{R}^3$ is the ground truth 3D point of regard (cf. Section 4.4.1). Notice that this strategy mainly evaluate errors on $\mathbf{v}_t$, rather than in $\mathbf{o}_t$. However, errors on the estimation of $\mathbf{o}_t$ have a lesser impact on predictions of a distant point of regard than errors in $\mathbf{v}_t$.

- **Mean and median angular gaze error**. Using the above per-sample errors, we can then compute statistics on the evaluation set **E**. The default one is the mean angular gaze error:

$$\epsilon^\circ = \frac{1}{|\mathbf{E}|} \sum_{t \in \mathbf{E}} {\epsilon^\circ}_t \tag{4.6}$$

although the median $\bar{\epsilon}^\circ$ can be useful in some situations $\bar{\epsilon}^\circ = \text{median}(\{{\epsilon^\circ}_t\}_{t \in \mathbf{E}})$.

4.5 Proposed benchmarks

The EYEDIAP database was designed such that each recording session could be well characterized in terms of participant, head pose activity, visual target and sensing conditions. The goal of this systematic collection was then to design experimental protocols, or benchmarks, which are suitable to evaluate a particular characteristic of a gaze estimation algorithm, while controlling for the rest of variables. Therefore, in this section we describe a few relevant proposed evaluation benchmarks. An additional benchmark is defined in Appendix A.2. These protocols are examples on how to design and conduct experiments with this database.

Notice this dataset has two main types of visual targets: 3D floating target (FT) and screen target (CS or DS). Therefore, the evaluation protocols are defined independently of the visual target which, once specified, lead to the definition of the actual recording sessions to be used. No restrictions are made on the modality used (RGB, RGB-D, HD, etc.), but in our experiments we mainly used the RGB-D data.

4.5.1 Benchmark 1: Gaze estimation accuracy

In this protocol, we evaluate the accuracy of a gaze estimation algorithm $\mathscr{H}$ under minimal variation of all parameters which are not gaze itself. Therefore, for a recording session Σ where the only variation is in the gaze direction itself (e.g., the session *1_A_DS_S*), we can design an experiment in which the training set **T** is defined as the first temporal half of Σ and the test set **D** is the second temporal half of Σ, such that **T** and **D** do not overlap.

The goal then consist on obtaining the mean gaze angular error ϵ° from **E**, where **E** is a subset of samples from **D**, defined as discussed in Section 4.4.6. The relevant sessions depend on the visual target type, as follows:

- **Screen target**: Includes all recording sessions with static head pose (S) and the screen target (either CS or DS). This makes 14 sessions in total (1 per participant), see Table 4.1.
- **3D floating target**: Includes all sessions with a static head pose (S) and the *FT* target, for any ambient condition A or B. This makes a total of 19 sessions.

The process of training, testing and evaluation is repeated for all relevant recording sessions and the mean angular error ϵ° is computed as the average among all the per session mean angular errors. Notice that, here, the intention is to obtain the method accuracy for conditions in which the test data is similar to the training data, i.e, same participant, minimal head pose variations and same sensing conditions.

4.5.2 Benchmark 2: Head pose invariance

In this case the objective is to measure how much does the gaze estimation accuracy decays when the participant perform changes in head pose. Two experiments are conducted per participant k, where the relevant recording sessions are defined from the desired visual target:

- **Experiment 1**. In this experiment, an evaluation of the gaze estimation accuracy is conducted for a static (S) head pose, such that the obtained mean angular error is $\epsilon^{\circ S}$. We denote the used recording session as Σ_S. This step follow exactly the procedure defined in Section 4.5.1.
- **Experiment 2**. In this case, we evaluate the method $\mathscr{H}$ in the presence of head pose variations (M) obtaining a mean angular error of $\epsilon^{\circ M}$. Notice that, for a recording session Σ_S used for "Experiment 1", there is an equivalent recording session Σ_M with the same configuration (participant, ambient conditions and visual target) except that it includes the case of head pose variations (e.g., Σ_S ="1_A_DS_S" and Σ_M ="1_A_DS_M"). Then, for this experiment, let the training set **T** be session Σ_S, whereas the test set **D** is Σ_M. **E** is then defined accordingly and the error $\epsilon^{\circ M}$ is computed over **E**.

The errors $\epsilon^{\circ S}$ and $\epsilon^{\circ M}$ are computed for every pair (S and M) of relevant recording sessions. As final result, the values $\epsilon^{\circ S}$ and $\epsilon^{\circ M}$ are averaged among all experimental pairs and reported.

Variants of this benchmark may include training data obtained from Σ_M, as well as from Σ_S; as long as the sets **T** and **D** are disjoint.

4.5.3 Benchmark 3: Person invariance

In this benchmark the goal is to evaluate how well does a method $\mathscr{H}$ generalize to unseen users. The set of relevant sessions are the same as for Benchmark 1 (Gaze estimation accuracy) but we discard the sessions from ambient conditions B (to avoid training and evaluating on the same user). Then two experiments will be conducted for a participant k as follows:

- **Experiment 1**. This follow exactly the methodology defined for Benchmark 1. The output is the gaze estimation error $\epsilon^{\circ k}$ obtained when there is a person specific training and the evaluation is done within the same recording session Σ_k (the session for user k). Notice that, as in Benchmark 1, the training and test sets are disjoint.

- **Experiment 2**. This experiment follows a leave-one-person-out cross validation scheme. Therefore, for a user k, we define the training set from the sessions from all other users as $\mathbf{T} = \cup_{j \neq k} \mathbf{T}_j$, where $\mathbf{T}_j$ corresponds to the entire session Σ_j, in which all other parameters (head pose, ambient conditions and visual target) are the same as for session Σ_k, except that it is the session corresponding to the participant j. The test set is the session specific to user k, i.e., session Σ_k. $\mathbf{E}$ is then defined accordingly and the error $\epsilon^{\circ \setminus k}$ is computed over $\mathbf{E}$.

Both experiments are conducted for all participants, the values $\epsilon^{\circ k}$ and $\epsilon^{\circ \setminus k}$ are computed per session and their average is finally reported.

4.6 Conclusion

In this Chapter we have described the EYEDIAP database, which we have collected and made publicly available to the research community. This dataset was designed for the development and evaluation of gaze estimation algorithms based or RGB or RGB-D data. We intended to address the need of the research community for standardized benchmarks, and to develop an experimental framework which can be used for the rest of this thesis.

The resulting database is rich and diverse. Furthermore, it is representative of a wide spectrum of applications and scenarios. Most variables which have an influence on gaze estimation algorithms based on remote sensors and natural illumination have been systematically isolated. This includes the head pose, person, ambient conditions and type of visual target. This allows the definition of specific benchmarks, or experimental protocols, which evaluate, in a controlled manner, the robustness of gaze estimation algorithms to any of these variables.

We therefore described in detail the recording methodology and the resulting set of recording sessions. Additional information was automatically extracted and provided to researchers, such as the calibration data of the sensors (both intrinsic and extrinsic ones), the head pose and eyes tracking results, etc. Finally, we provided examples on experimentals protocols designed to evaluate the robustness of gaze estimation algorithms towards any of the aforementioned variables.

We believe this database is of high value to researchers as it will help to advance the development of gaze estimation algorithms for diverse scenarios, including those with less constrained conditions. This data has been crucial for the development and validation of the contributions of this thesis, in particular, for the evaluations presented in Chapter 5, and Chapter 6.

5 Appearance Based Gaze Estimation

5.1 Introduction

In this chapter we address the problem of appearance based gaze estimation from remote RGB-D sensors. The main goal is to develop a system capable of gaze sensing under minimal or even under unexistent user cooperation. Many applications, within the fields of human-robot, human-human and human-computer interaction (respectively HRI, HHI and HCI), may greatly profit from such system. In these scenarios, remote sensors with a large field of view are needed to minimize user cooperation, and to accommodate unconstrained user movements. Nevertheless, these conditions normally lead to the problem of low resolution sensing of the eye image, as shown in Figure 5.1.

Appearance based gaze estimation methods have the potential to address low resolution sensing conditions, as they do not rely on the extraction of local eye features, such as the iris/pupil center or corneal reflections. Instead, these methods learn a direct mapping from the eye image appearance to the gaze parameters. In the past, many approaches have been proposed which fall within the appearance based gaze estimation paradigm (see Section 2.2.2). However, these methods have problems to generalize. Therefore, they are normally restricted to head mounted setups and/or well controlled laboratory conditions.

The challenge consists of learning a mapping invariant to the many elements which influence the appearance of the eye image, such as the user, head pose or viewpoint, illumination conditions, image resolution or eye distance to the sensor, contrast, eyelids shape and movements, specular reflections, motion blur, self occlusions, etc.

Few methods have addressed simultaneously the problems of head pose and user invariance[1]. Moreover, to our knowledge, no previous works relying on the appearance based framework address the problem of gaze estimation within a wide 3D space, which would allow the usage of gaze sensing for tasks within diverse HRI, HHI or HCI scenarios, in contrast to the traditional screen gazing case.

[1] Only very recent proposals, see Chapter 2 for a detailed literature review.

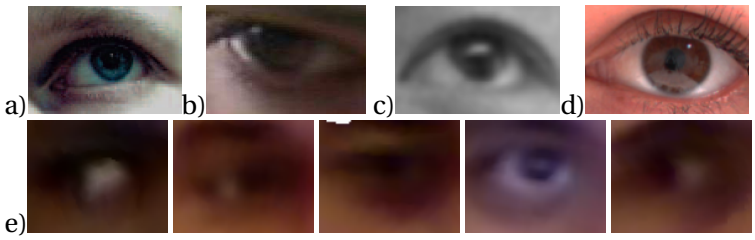


Figure 5.1: Sample eye images taken from a) Martinez et al. [2012] b) Noris et al. [2010] c) Lu et al. [2011a] d) Schneider et al. [2014]. e) Samples Kinect-based eye images from the EYEDIAP database [Funes Mora et al., 2014a], taken at ≈ 1.2m; Notice the much poorer resolution and contrast in the latter due to the capture from remote consumer sensors.

Therefore, in this chapter we address the challenges of gaze estimation in the 3D space under varying head poses and users. Diverse contributions are proposed to address these problems:

- **Head pose invariant 3D gaze estimation from consumer RGB-D sensors** (Section 5.2). We propose a methodology which profits from consumer RGB-D sensors to rectify the eye image appearance into a canonical head pose viewpoint. This framework effectively brings head pose invariance into existent appearance based gaze estimation methods, as we demonstrate empirically relying on recent appearance based methods. The methodology is suitable to address the continuous space of head poses and it does not require additional training data. To our knowledge, only the method from Valenti et al. [2012] relies on a similar methodology for gaze estimation. However, their pose-rectification approach is based on a cylindrical head model, which introduces large distortions, inadequate for appearance based methods and, as it is based on a monocular setup, it suffers from depth/scale ambiguity. Our approach was first proposed in [Funes Mora and Odobez, 2012] and it was further developed in [Funes Mora and Odobez, 2015].
- **Coupled adaptive linear regression** (Section 5.3.4). Within a sparse linear reconstruction gaze regression framework, we propose to introduce anatomical constraints which couple the gaze direction of the left and right eyes. This results on more accurate and robust gaze estimation. This approach was published in [Funes Mora and Odobez, 2013].
- **Person invariant gaze estimation** (Section 5.4). Aiming to minimize user cooperation, we investigate the performance of recent appearance based gaze estimation when creating person invariant models. We use the EYEDIAP database for the training and evaluation of these models. In addition, as published in [Funes Mora and Odobez, 2013], we propose an unsupervised model selection mechanism based on the sparse linear reconstruction of test samples from a pool of person dependent gaze appearance models. This results on an adequate trade-off between gaze estimation accuracy and computational cost.
- **Inter-user eye alignment** (Section 5.4.2). We address the eye image alignment problem with arises in the context of person invariant gaze estimation. Contrary to prior works which rely on facial features such as eye corners, we propose a method called synchronized delaunay implicit parametric alignment, which relies on the joint registration of gaze synchronized eye images. This approach effectively increases the accuracy of person invariant gaze estimation models.

Extensive experiments were conducted to validate these contributions and promising results are obtained for the EYEDIAP database, despite the low resolution of the input eye images, large range of gaze directions and significant head pose variations.

This chapter is structured as follows. The head pose invariant appearance based gaze estimation framework is described in Section 5.2. Gaze appearance based methods, suitable for this context, are described in Section 5.3. Extensions to acquire user invariance are described in Section 5.4. The gaze estimation data and experimental protocols are presented in Sections 5.5, followed by results in Section 5.6. Section 5.7 discusses limitations and future work. Finally, Section 5.8 concludes this chapter.

5.2 Head pose invariant gaze estimation

In this section we describe our 3D rectification methodology which allows for head pose invariant appearance based gaze estimation. We first introduce the overall approach, and then detail the different steps involved in the rectification process.

5.2.1 Approach overview

The main principle of this approach is to rectify the eye images into a canonical (frontal) head viewpoint and scale regardless of the actual head pose by exploiting the calibrated RGB-D input data, and then estimate the gaze in this canonical view.

The different steps involved in this process are depicted in Fig. 5.2. First, we assume a user specific 3D face model is available. In this Chapter we will assume this model is learned in an offline step. Then, in the online phase, the proposed method consists of the following steps:

1. At each time step t, the 3D head pose $\mathbf{p}_t$ is estimated.
2. The face region is rectified into a frontal view from the input RGB-D data and the estimated head pose, leading to a rendered image $\mathbf{I}^R$ for each eye. An eye alignment step is then applied in order to crop the eye region $\mathbf{I}^C$.
3. The gaze direction $\mathbf{v}^{\mathbf{h}}$ in the head coordinate system is estimated from $\mathbf{I}^C$.
4. The obtained gaze is mapped back into the world coordinate system (**WCS**) using the pose $\mathbf{p}_t$, and used along with the eyeball center $\mathbf{o}^{\text{WCS}}$ to define the gaze line of sight (LoS).

In the following, we describe the aforementioned steps in detail.

5.2.2 3D Head pose and eyes tracking

Our gaze estimation methodology relies on the accurate tracking of the head pose. To this end, we employ the approach which has been described in detail in Chapter 3, Section 3.3. This

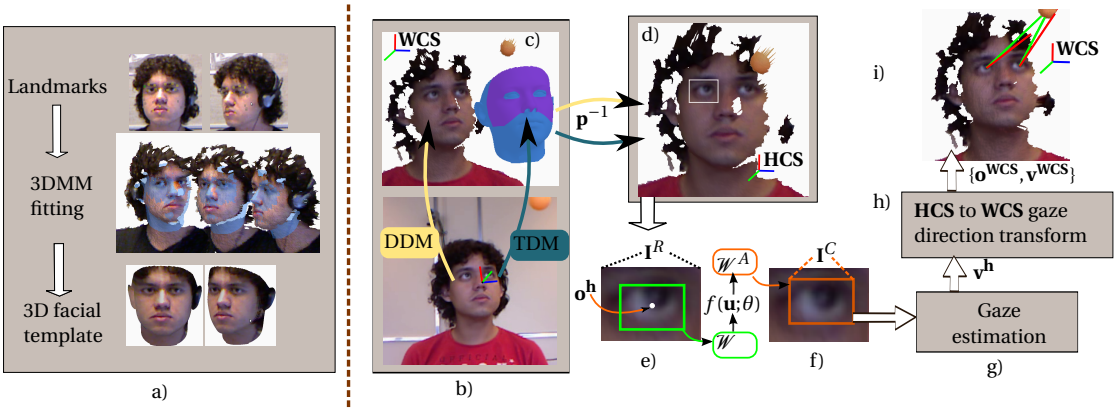


Figure 5.2: Proposed method pipeline. a) Offline step. From multiple 3D face instances, the 3DMM is fit to obtain a person specific 3D model. b-i) Online steps. b) The 3DMM fitted face model is registered at each instant to the depth data of the RGB to obtain the head pose (the region used for tracking is rendered in purple in the 3DMM mesh in c). c) a 3D textured mesh is obtained by binding the RGB image either to the depth D channel of the sensor (shown Data Driven Mesh, *DDM*), or to the 3D facial template (template-driven mesh, *TDM*; note: only the template is shown). d) the textured mesh is rendered in a frontal pose by rotating it using the inverse head pose parameters, for which an eye image region $\mathbf{I}^R$ can be obtained. e-f) as a predefined region $\mathcal{W}$ around the eyeball center $\mathbf{o}^\mathbf{h}$ may not consistently crop the same eye part across users, an alignment warping f learned for each user is applied to $\mathcal{W}$ and defines the region $\mathcal{W}^A$ where the image should be cropped. g-i) the gaze $\mathbf{v}^\mathbf{h}$ in the head coordinate system **HCS** is estimated from the cropped image $\mathbf{I}^C$, and then transformed back in the **WCS** to obtain the line of sight (green line, ground truth; red line, estimated LoS).

method relies on the 3D registration of a person specific face model to depth data (represented as a 3D mesh). To account for non-rigid face deformations, e.g., when people speak, we fit only the upper part of the face, as shown in Fig. 5.2c. The registration is done using the iterative closest points algorithm based on a point-to-plane cost formulation. Therefore, for an input frame at time t, the output of the head pose tracking algorithm are the pose parameters $\mathbf{p}_t = \{\mathbf{R}_t, \mathbf{t}_t\}$, of a rotation $\mathbf{R}_t \in \mathbb{R}^{3\times3}$ and a translation $\mathbf{t}_t \in \mathbb{R}^3$.

In this Chapter, the person specific face model will be assumed to be obtained offline, by fitting a 3D Morphable Model (3DMM) to a set of facial landmarks annotated RGB-D snapshots of the subject's face, as seen in Fig. 5.2a. The 3DMM fitting methodology is described in Chapter 3, Section 3.2. From the 3DMM topology we can extract an adequate approximation of the subject's eyeball center $\mathbf{o}^\mathbf{h}$, defined with respect to the head coordinate system (**HCS**). This is computed as the average of the eyelids contour points of the 3DMM fitted person specific model (whose indices are known from the 3DMM topology), which is then translated 5mm in depth towards the head interior.

Finally, we can then compute the frame by frame -approximate- eyeball center in 3D, with

respect to the **WCS** based on $\mathbf{o^h}$ and the head pose obtained at time t, as follows:

$$\mathbf{o}_t^{\text{WCS}} = \mathbf{R}_t \mathbf{o^h} + \mathbf{t}_t \tag{5.1}$$

To avoid repetitions, the descriptions here focus on a single eye, but the process of eye tracking and gaze estimation should be understood as done for both the left and right eye separately.

5.2.3 Eye appearance rectification

The key step for head pose invariance is the rectification of the face texture to a canonical head pose viewpoint, which is done as follows. Given a textured 3D mesh (i.e., a mesh where each 3D point is associated with an RGB color) of the face image at time t, we render it after applying the rigid transformation $\mathbf{p}_t^{-1} = \{\mathbf{R}_t^\top, -\mathbf{R}_t^\top \mathbf{t}\}$, i.e., the inverse of the estimated head pose, generating a frontal-looking face image (Fig. 5.2d).

As textured mesh, we considered two possibilities. A **data-driven mesh (*DDM*)**, obtained by mapping the RGB texture to the raw depth mesh built from the D channel of the sensor. And a **template-driven mesh (*TDM*)**, resulting from the mapping of the texture to the person specific 3D face model fitted to the data. Note that the rectification does not require a prior knowledge of the user's appearance and only assumes that the calibration is accurate enough to bind the RGB data to a mesh surface.

Both methods have their pros and cons (see Fig. 5.8 for rectification samples). We could expect a better accuracy from the *DDM*, but this is subject to all types of sensor noise from the depth channel like the measurement noise or the absence of data due to sensing issues (e.g., when being too close to the sensor, see Experimental Section). The template approach, depending on the 3DMM fitting quality, provides a looser fit to the actual user eye 3D surface, but provides a smoother surface for the rectification and frontal rendering.

5.2.4 Eye image alignment

This step is illustrated in Fig. 5.2e-f). Thanks to the rectification, we can extract an image $\mathbf{I}^R$ around the eye region, out of which a more precise eye image could be extracted within a predefined window $\mathcal{W}$ whose position is defined by the eyeball center $\mathbf{o^h}$.

In principle, due to the 3DMM fitting, this window should capture the same part of the eye for different users, if head pose tracking errors are not considered. However, due to the uncertainty affecting the accuracy of the 3DMM fitting, or the natural human variations in the eyeball localization, which are not perfectly correlated to the position of facial features (e.g., the eye corners), this may not be the case, as illustrated in Fig. 5.14.

To address this issue, the parameters θ of an alignment transform are learned for each user using a small set of samples, as explained more precisely in Section 5.4.2. They are used to transform the window $\mathcal{W}$ into the aligned one $\mathcal{W}^A$, defining the region of $\mathbf{I}^R$ where the image $\mathbf{I}^C$ is actually cropped for further processing.

Note that the coordinate transformations in the $\mathbf{I}^R$ image domain can be directly reinterpreted within the **HCS** domain. Therefore, the alignment transform can be seen as a local transformation of the 3DMM fitted model itself, as a refinement step. Indeed, when θ defines a translation, and assuming the estimated head pose is not affected by such 3DMM local refinement, the refined face model would generate the same eye image to $\mathbf{I}^C$, in particular for the *DDM* case

5.2.5 Gaze estimation

The pose-rectified and aligned cropped eye image $\mathbf{I}^C$ is used to estimate the gaze direction using a regression estimator. As these images are normalized, any standard method can be used. As we mentioned, in this Chapter we focus on recent appearance based methods (ABMs), which are described in more detail in Section 5.3.

The input to the gaze estimator is the image $\mathbf{I}^C$ and the output is the gaze direction, parametrized by the gaze yaw and pitch angles, or equivalently, by the unitary 3D vector $\mathbf{v}^{\mathbf{h}} \in \mathbb{R}^3$ defined in the head coordinate system (**HCS**). This vector can be transformed into the **WCS** system and used with the eye center $\mathbf{o}^{\mathbf{WCS}}$ to define the line of sight (3D ray in the **WCS**) as:

$$LoS^{\mathbf{WCS}}(l) = \mathbf{o}^{\mathbf{WCS}} + l\,\mathbf{v}^{\mathbf{WCS}}, \tag{5.2}$$

where $\mathbf{v}^{\mathbf{WCS}} = \mathbf{R}\mathbf{v}^{\mathbf{h}}$, $\mathbf{R}$ is the head rotation and $l \in [0, \infty[$.

5.3 Appearance based gaze estimation methods

Thanks to the head pose rectification and alignment steps, the gaze estimation problem is simplified and we can apply any method that was originally designed for a fixed head pose based setup, or for head mounted cameras.

We assume we are given a training set $\mathbf{T} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ containing N pairs of descriptors $\mathbf{x}_i \in \mathbb{R}^D$ (extracted from the cropped images $\mathbf{I}^C$), and associated gaze directions $\mathbf{y}_i \in \mathbb{R}^2$ represented by their gaze yaw and elevation (or pitch) angles. We also define $\mathbf{X} \in \mathbb{R}^{D\times N}$ (resp. $\mathbf{Y} \in \mathbb{R}^{2\times N}$) as the matrix where each column contains one descriptor (resp. gaze direction) from $\mathbf{T}$. The goal is to infer the gaze direction $\hat{\mathbf{y}}$ for a test sample $\hat{\mathbf{x}}$. Conceptually, the appearance based gaze estimation methodology is depicted in Figure 5.3.

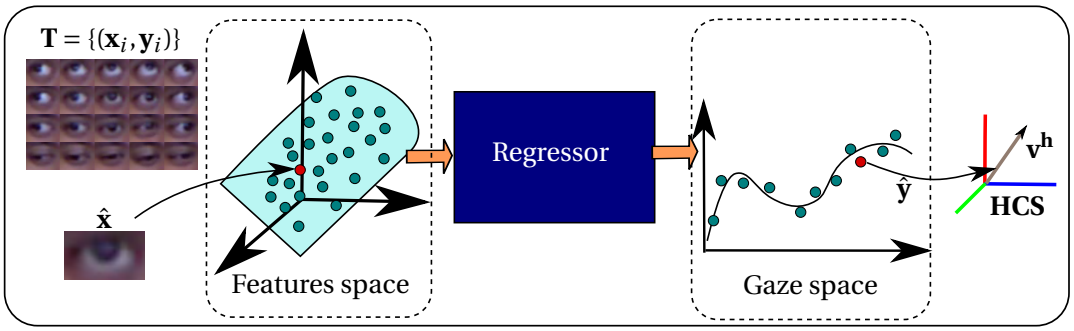


Figure 5.3: Appearance based gaze estimation methodology. We assume a training set **T** is available. A test image, represented by its descriptor $\hat{\mathbf{x}}$, is mapped directly into gaze parameters by a regressor, which was trained based on the set **T**. In our framework the gaze parameters are the gaze yaw and pitch angles of the unitary vector $\mathbf{v}^{\mathbf{h}}$, defined with respect to the **HCS**.

In the rest of this section, we focus on a baseline (kNN) and the recent state-of-the-art methods [Lu et al., 2011a, Noris et al., 2010, Martinez et al., 2012, Funes Mora and Odobez, 2013] that have shown good performance and that we have implemented.

5.3.1 k-Nearest Neighbors (kNN)

Features. The eye image[2] $\mathbf{I}^C$ is first contrast normalized (by setting their mean to 128 and normalizing their standard deviation to 40), and all pixels are stacked into a column vector to form the descriptor $\mathbf{x}$.

Regression. The $K = 5$ nearest neighbors of the test sample $\hat{\mathbf{x}}$ (according to the euclidian distance) are extracted, and their gaze directions $\{\mathbf{y}_k\}$ are used to compute the gaze of $\hat{\mathbf{x}}$ as

$$\hat{\mathbf{y}} = \sum_{k \in \mathcal{K}} w_k \mathbf{y}_k, \tag{5.3}$$

where $\mathcal{K}$ contains the neighbors indices, and the weights $\{w_k\}$ are set inversely proportional to the distance to the test sample.

5.3.2 Multi-level HoG and Retinex Support Vector Regression (H-SVR and R-SVR)

These two methods were proposed by Martinez et al. [2012] and Noris et al. [2010] for head mounted camera systems which required some invariance to illumination, and differ only on the feature type: Multi-level HoG (mHoG) features for the former, retinex for the latter.

mHoG Features. The image is divided into 1×2, 3×1, 3×2 and 6×4 block regions, each of

[2] Note that in all methods $\mathbf{I}^C$ is a gray-scale image of size 55×35. This is a conservative choice, since in our experiments eye image sizes almost never go beyond $\approx 20 \times 15$. It should however not be harmful in principle.

which is divided into 2×2 cells from which signed HoG histograms of 9 orientations [Dalal and Triggs, 2005] are computed (see Fig. 5.4). The histograms are L2-normalized per block. Then $\mathbf{x}$ corresponds to all concatenated HoG histograms. Gradient features can provide robustness against illuminations issues, while histograms may lead to more robust features against noisy location of the eye region. Indeed, in the study of Schneider et al. [2014] (on rather high resolution images), a comparison with 7 other features showed that multilevel HoG was performing best[3], with SVR (out of 6 classifiers) being the best regressor.

Retinex Features. To minimize the impact of non-uniform eye illumination variations, a retinex technique, weighted according to local contrast [Choi et al., 2007], is applied to the input image $\mathbf{I}^C$. The image pixels are then stacked in column to generate $\mathbf{x}$. Note that this feature was not tested (and thus compared with mHoG) in [Schneider et al., 2014].

Regression. The regression of the gaze parameters is done using a ν-Support Vector Regression (νSVR), where each gaze angle is regressed separately.

The principle of SVR is to learn a linear regression function in a high dimensional space where the input features have been implicitly mapped, and in which the inner product between two elements i and j can equivalently be computed as $\mathbf{k}(\mathbf{x}_i, \mathbf{x}_j)$, i.e., the kernel value between the elements in their original space. The parameters are then obtained by optimizing the structural risk, allowing to find a compromise between overfitting and model complexity. As in [Martinez et al., 2012, Noris et al., 2010], we rely on the νSVR rather than ϵSVR in order to have a better control of the learning error. More details can be found in [Smola and Schölkopf, 2004].

The hyper-parameters of the models are $\mathbf{C}$ and ν, which control the weights of the different costs of the objective function, and the precision γ of the Radial Basis function kernel $\mathbf{k}$ that we use. For a given experiments, these parameters were set through a 10-fold cross validation on the training data with a grid search over reasonable values.

5.3.3 Adaptive Linear Regression (ALR)

This method was proposed by Lu et al. [2011a], which we will describe as follows:

Features. The $\mathbf{I}^C$ image is first contrast-normalized as in the kNN case. The descriptor $\mathbf{x} \in \mathbb{R}^{15}$ is then created by dividing the image into 5×3 regions, and computing the cumulative intensity in each region (cf. Fig. 5.4). To gain further robustness against illumination changes, the resulting values are normalized such that $\mathbf{1}^\top \mathbf{x} = 1$, where $\mathbf{1} = [1, 1, \ldots, 1]^\top$.

Regression. Estimation is formulated as a sparse reconstruction of the test sample $\hat{\mathbf{x}}$ from a linear combination (represented by $\mathbf{w}$) of the training samples $\{\mathbf{x}_i\}$. The optimal weights $\hat{\mathbf{w}}$ are

[3] Except when combined with local binary patterns, although the gain in accuracy was negligible: 0.02°

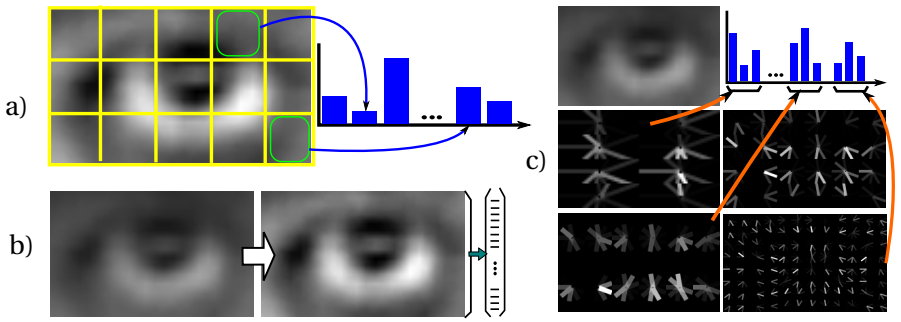


Figure 5.4: Features extraction. a) Descriptor used for Adaptive Linear Regression (ALR) and Coupled Adaptive Linear Regression (CALR) b) Weighted retinex c) multilevel HoG .

obtained by solving:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \|\mathbf{w}\|_1 \qquad s.t. \qquad ||\mathbf{X}\mathbf{w} - \hat{\mathbf{x}}||_2 < \epsilon, \tag{5.4}$$

and then used to compute the test sample's gaze as $\hat{\mathbf{y}} = \mathbf{Y}\hat{\mathbf{w}}$. The implicit assumption is that enforcing sparsity will induce the selection of only a few samples within a small region of the appearance manifold, such that the same linear mapping in the appearance and gaze spaces can be exploited.

In the above formulation, the parameter ϵ plays an important role. Lu et al. [2011a] recommended to obtain ϵ from cross validation on the training set. However, our much noisier data drastically differ from the well controlled conditions used by Lu et al. [2011a]. Therefore the ϵ value resulting from cross validation usually happened to be too restrictive at test time. We thus resorted to the original proposition by the same authors, where the optimal value of ϵ should be determined when the resulting $\|\mathbf{w}\|_1$ is equal to 1. In practice, we evaluated this using seven predefined values of ϵ, at the cost of longer computation time.

Finally, note as well that solving the problem in Eq. 5.4 is difficult, with a computation complexity increasing rapidly with respect to the number of training samples, thus limiting its application to small training sets. Nevertheless this was shown sufficient to obtain good accuracy.

5.3.4 Coupled Adaptive Linear Regression (CALR)

We proposed the CALR method in [Funes Mora and Odobez, 2013].

Features. The features are the same as in Section 5.3.3.

Regression. The method described in Section 5.3.3, i.e., Adaptive Linear Regression, is applicable to the left ("l") and right ("r") eye to obtain their gaze directions separately. However, it is known that both eyes fixate jointly a single 3D point. We therefore proposed to extend ALR to

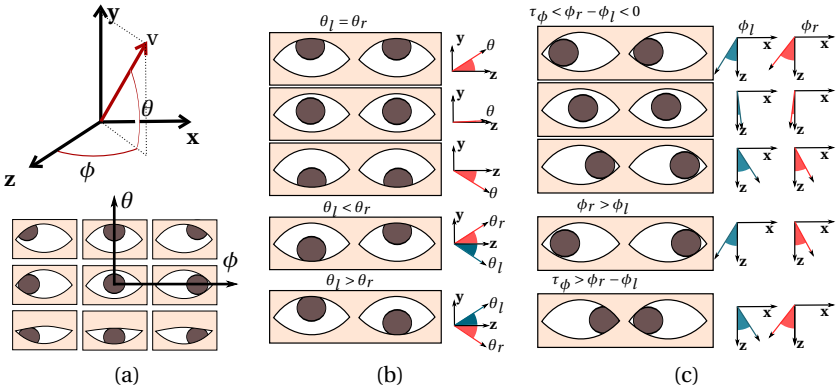


Figure 5.5: Coupled adaptive linear regression angle constraints. (a) Gaze parametrization in the **HCS**. (b) Gaze elevation constraints. (c) Gaze yaw constraints. For both cases ((b) and (c)) we show the 3 examples following the constraints and 2 examples which break the constraints.

build upon this fact. Let us first denote the gaze direction angles, defined with respect to the **HCS**, as $\mathbf{g^h} := \{\phi, \theta\}$, as shown in Figure 5.5a.

Let us denote the gaze values for the left and right eyes as $\{\mathbf{g}_l^\mathbf{h}, \mathbf{g}_r^\mathbf{h}\} = \{\{\phi_l, \theta_l\}, \{\phi_r, \theta_r\}\}$, we can then incorporate appropriate constraints into the ALR problem as follows. As a first observation, notice that if the eyes are horizontally aligned, then their gaze elevation should be the same, i.e., $\theta_l = \theta_l$ as shown in Fig. 5.5b. This allows us to represent the gaze elevation, for both eyes, as a single parameter θ. Secondly, we can observe that since the two eyes are assumed to fixate at a single 3D point, the condition $\phi_r < \phi_l$ should be satistied, which accounts for all amounts of eye vergence. Equality occurs when gazing a point at an infinite distance. We can also limit the distance at which the closest 3D point is expected to be by setting $\phi_r - \phi_l > \tau_\phi$, where τ_ϕ is a constant. This is illustrated in Fig. 5.5c.

We formalize these observations by estimating the left and right gaze directions jointly by solving the following constrained optimization problem:

$$\begin{aligned} \hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \|\mathbf{w}\|_1 \qquad s.t. \qquad & ||\mathbf{X}\mathbf{w} - \hat{\mathbf{x}}||_2 < \epsilon, \\ & \mathbf{Y}_{\theta_l}^\top \mathbf{w}_l - \mathbf{Y}_{\theta_r}^\top \mathbf{w}_r = 0, \\ \tau_\phi < & \mathbf{Y}_{\phi_r}^\top \mathbf{w}_r - \mathbf{Y}_{\phi_l}^\top \mathbf{w}_l < 0, \end{aligned} \tag{5.5}$$

where we redefine $\mathbf{w} = [\mathbf{w}_l^\top, \mathbf{w}_r^\top]^\top$, $\hat{\mathbf{x}} = [\hat{\mathbf{x}}_l^\top, \hat{\mathbf{x}}_r^\top]^\top$, and $\mathbf{X}$ as the following block matrix:

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_l & \mathbf{0} \\ \mathbf{0} & \mathbf{X}_r \end{bmatrix}, \tag{5.6}$$

in which $\mathbf{0}$ is a zero filling matrix. Note that $\mathbf{Y} := [\mathbf{Y}_\phi, \mathbf{Y}_\theta]$, i.e., the gaze parameters are defined in terms of the gaze elevation and yaw angles. Eq. 5.5 can be solved as a Second Order Cone Programming (SOCP) problem.

5.3.5 Head pose (HP)

Finally, we include a"dummy" algorithm, which is denoted "Head pose" (HP). It corresponds to not using a gaze estimator, but always setting the gaze parameters to zero, or equivalently, $\mathbf{v}^{\mathbf{h}} = [0,0,1]^\top$ (i.e., gazing forward). In the 3D space, this corresponds to using the head pose as gaze direction. This strategy is reported in the experimental section to convey the actual amount of gaze variations observed within our data.

5.4 Person invariant gaze estimation

In this section we address the person invariance problem, which we denote as the situation in which there is no training data available for the test subject in order to learn an appearance to gaze regression model.

In Section 5.4.1 we describe how we learn person invariant classifiers for the different gaze models of Section 5.3. Then, in Section 5.4.2, we address the cross-user eye image alignment problem.

5.4.1 Person invariant classifier

In this section we consider two main strategies to generate a suitable person invariant model.

Joint model training

We assume that a dataset $\mathbf{T}_i$ of gaze annotated training samples processed according to the method outlined in Section 5.2.1 (Fig. 5.2) is available for each of the M subjects. The simplest strategy to acquire person invariance is to create a joint training set $\hat{\mathbf{T}} = \cup_{i=1}^{M} \mathbf{T}_i$; an approach that can immediately be applied to the kNN, R-SVR and H-SVR classifiers.

Unsupervised model selection

As mentioned in Section 5.3.3, an important limitation of ALR is its computation time, which prohibits the usage of $\hat{\mathbf{T}} = \cup_{i=1}^{M} \mathbf{T}_i$ as training data. To address the person invariance case, we instead propose an unsupervised selection of person-specific training sets within the database. This approach was proposed in [Funes Mora and Odobez, 2013].

It is based on the observation that, when using $\hat{\mathbf{T}}$ to solve Equation 5.4, we can compute the weight given to each person i as $W_i = \sum_{j|j\in\mathbf{T}_i} |w_j|$, where $\{w_j\}$ are weights obtained for each

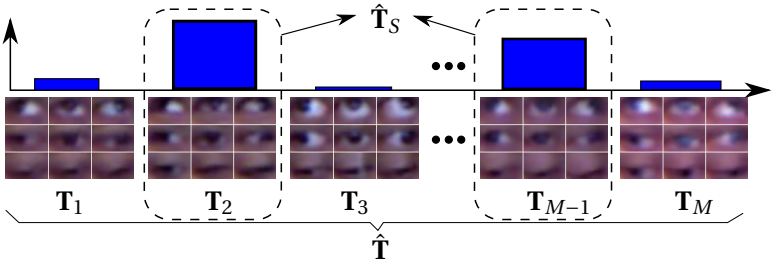


Figure 5.6: Unsupervised model selection based on adaptive linear regression. The set $\hat{\mathbf{T}}$ is composed of models obtained from different people $\{\mathbf{T}_i\}$. Based on the accumulated sparse reconstruction weights distribution, we can select a subset of person specific models $\hat{\mathbf{T}}_S$ adequate for gaze estimation for the test subject, thus avoiding to use the entire set $\hat{\mathbf{T}}$.

sample after solving Equation 5.4. If the test image radically differs from the samples of person j, then it is expected for W_j to tend to zero, due to sparsity. Whereas, if the eyes appearance between two people is similar, then W_i will be high. By accumulating the W_i's through time, we can rank the models according to their relevance to the test person, shown in Figure 5.6. More precisely, we create the selected model $\hat{\mathbf{T}}_S$ as the subset of models with the highest weights. In practice, a small subset of the test samples of the given user are used and processed with this approach using the full model $\hat{\mathbf{T}}$. Notice that this process can be computationally demanding. Nevertheless, once the models are selected, the estimation for the rest of the samples in the test set can be done much faster.

5.4.2 Alignment

One issue when combining data from different users is that the image cropping defined from the proposed 3D rectification may not extract exactly the same eye region. For instance, the data collected for two users may exhibit a systematic translation bias: roughly speaking, for the same gaze, in the cropped images, the iris location of the first user is systematically displaced by a few pixels from the iris location of the second user.

In practice, this spatial alignment error can result in a systematic gaze angular error bias when inferring the gaze of the first user using the training data from the second user. In the next subsections, we first present a standard approach to address the *alignment problem*[4], and introduce our proposed alignment methodology

Eye corner based alignment

To align eye images, the common strategy consists of locating the eye corners in a few frames, and use this information to estimate the parameters of the transformation that bring them back to a canonical position. The eye corner localization is often done manually (e.g., Martinez et al.

[4]Note that when the test data corresponds to the same subject as the one in the training set, the alignment is not needed, as we expect the cropping to be consistent between the test and training data.

[2012]), and then the same parameters used for all frames. Automatic eye corner localization methods have been used, but so far on high resolution images (e.g., see eye in Fig. 5.1d). For much lower resolution conditions such as in our data (see eyes in Fig. 5.1e) this can be problematic in terms of localization accuracy, despite important recent advancements (e.g., see Kazemi and Sullivan [2014]).

Besides the localization issue, we argue that this alignment strategy is not the optimal for the task of gaze estimation, as discussed below. We therefore present an alternative in the next section.

Synchronized Delaunay Implicit Parametric Alignment (SDIPA)

Ideally, an alignment strategy aiming at person invariance should be based on aligning the eyeball positions of the different subjects, and not necessarily specific facial features such as eye corners. However, as the eyeball centers are not directly observable, we propose instead to use a direct image registration technique. In this manner, the important eye structures (and in particular the iris) of different subjects gazing in the same direction are always located at the same place.

Alignment modeling. Assume we are given a set of training images $\mathbf{T}_i = \{(\mathbf{I}_k^i, \mathbf{y}_k^i), k = 1, ..., K_i\}$ for each user i. Our aim is to find for each user the parameters θ_i of a warping function $f(\mathbf{u};\theta_i)$ registering the input images into a canonical frame. More precisely, if $\mathbf{u}$ denotes the pixel coordinates in the canonical frame, the aligned images $\tilde{\mathbf{I}}_k^i$ are then defined as

$$\tilde{\mathbf{I}}_k^i(\mathbf{u};\theta_i) = \mathbf{I}_k^i(f(\mathbf{u};\theta_i)). \tag{5.7}$$

To compute the parameters $\Theta := \{\theta_i\}_{i=1}^M$, we make the assumption that *when two subjects gaze in the same direction*, their aligned images (particularly the iris region) should match and their intensity difference should be minimal. Note that while this might not necessarily hold for all gaze directions and pairs of people, due to iris color differences and the angular deviation of the visual axis, we expect this assumption to be valid on average, i.e., when considering a large number of people and gaze values to constrain the parameters estimation.

However, given the image $\mathbf{I}_k^i$ of subject i with gaze $\mathbf{y}_k^i$, it is unlikely to find an image with the same gaze in $\mathbf{T}_j$. To address this problem, we propose for each $\mathbf{I}_k^i$ to use $\mathbf{T}_j$ to synthesize (as described later in this section) an image (denoted $\mathbf{I}_{\mathbf{y}_k^i}^j$) for suject j with the same gaze direction. Based on the above assumption, the alignment problem can now be defined as minimizing

$$E(\Theta) = \sum_{i=1}^{M} \sum_{k=1}^{K_i} \sum_{\substack{j=1 \\ j \neq i}}^{M} \|\tilde{\mathbf{I}}_k^i(\cdot;\theta_i) - \tilde{\mathbf{I}}_{\mathbf{y}_k^i}^j(\cdot;\theta_j)\|_2^2 + \rho R(\Theta) \tag{5.8}$$

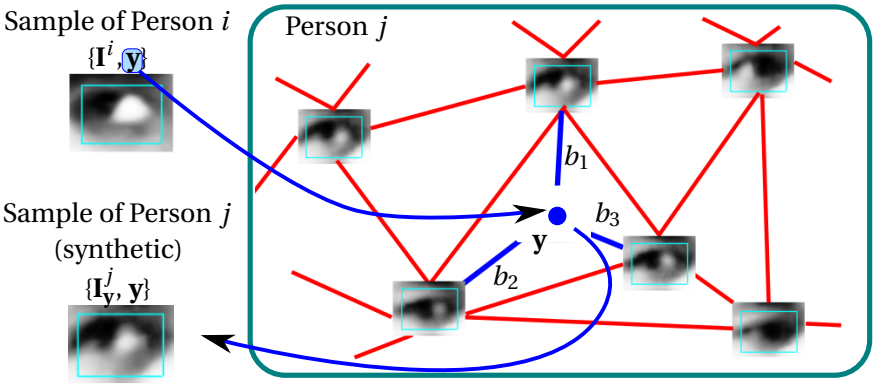


Figure 5.7: Synchronized Delaunay Interpolation used to establish eye image pairs with the same gaze direction for subjects i and j.

where $R(\Theta) = \sum_i^M \|\theta_i - \theta^{Id}\|_2^2$ is a regularization term[5] fostering the estimation of parameters close to those of the identity warp (i.e., θ^{Id} satisfies $\mathbf{u} = f(\mathbf{u};\theta^{Id})$).

To optimize Eq. 5.8, we use the first order Taylor series expansion and iteratively solve for changes on the parameters from the current estimate. For efficiency, we follow a similar strategy as proposed by Hager and Belhumeur [1998]. In the experiments presented in this thesis, we focus on the case in which the warping f represents a translation (of vector $\theta \in \mathbb{R}^2$).

Synchronized Image Synthesis. The aim is to be able to generate an eye image for any gaze parameters $\mathbf{y}$ using the training set $\mathbf{T}_j$ of subject j.

The process is illustrated in Fig. 5.7. In brief, we build a delaunay triangulation from the gaze angles $\{\mathbf{y}^j_k\}$ set (2-dimensional), and then find the set of vertices $\mathbf{S}^j(\mathbf{y})$ defining the triangle within which $\mathbf{y}$ falls, and generate the new image as:

$$\mathbf{I}^j_{\mathbf{y}} = \sum_{l \in \mathbf{S}^j(\mathbf{y})} b_l(\mathbf{y}) \mathbf{I}^j_l, \tag{5.9}$$

where b_l denotes the barycentric cooordinates of $\mathbf{y}$ in the triangle.

Alignment Procedure. We call the method described by Eq. 5.8 Synchronized Delaunay Implicit Parametric Alignment (SDIPA). In the thesis, we have exploited it to address two related tasks:

1. *Person invariant gaze model training.* In this task, the goal is to align a gaze annotated training set comprising different subjects, prior to learning the gaze regression models. We expect that exploiting aligned data will result in more accurate models. This is

[5] The direct minimization of the data term is ill-posed, as the same arbitrary transform applied to all the subjects generate the same error. In practice, we used a small value of ρ to make the optimization well-posed.

achieved by optimizing Eq. 5.8.

2. *Eye image alignment for a test subject.* The eye gaze model learned using the above alignment method (task 1) is person invariant, and can readily be applied to any new test subject. However, in some situations (as the example which will be described in Chapter 7), there is the possibility to gather for a test user a few samples with gaze information (e.g., a person looking at known location like another person, or simply, looking at the camera) that can be further exploited to improve the result. In this case, the same method can be used to find the eye alignment of this user with respect to the already aligned training set using these gaze annotated samples. This is simply done by adapting Eq. 5.8 and conducting the optimization only with respect to the parameters of a single subject (e.g., the θ_j of subject j considered as our test subject) while the other $\{\theta_i\}_{i\neq j}$ remain fixed. This second case can be seen as an adaptation step that is highly valuable in HRI and HHI scenarios. Notice that, even if conducting a proper gaze model training session is not possible in such scenarios, it might still be feasible to detect in a supervised or unsupervised manner instants at which the subject is fixating a given (known) target. These instances can be used to collect the few samples needed to find the test subject's alignment.

5.5 Experiments

In this section we first provide more details on our gaze estimation system implementation. In Sections 5.5.2 and 5.5.3 we describe the data and protocol we followed to conduct our experiments on gaze estimation.

5.5.1 Implementation details and speed

Details regarding the 3DMM fitting and the head pose tracking speed can be found in Section 3.5.1. In this Section we will describe details regarding the gaze specific elements.

Eye image rectification

The rectification process is implemented as an OpenGL rendering. As part of the OpenGL rendering instructions, the 3D scene is rigidly transformed according to the inverse head pose. The rendering is then defined as an orthographic projection.

Alignment

The warping function f used in our experiments is a translation ($f(\mathbf{u};\theta) := \mathbf{u}+\theta|\theta \in \mathbb{R}^2$)) which we found sufficient to (implicitly) align the eyeball position across subjects after rectification.

Person invariant gaze model training: solving Eq. 5.8 to find the per-subject eye alignment

parameters prior to the training of a person invariant model can take 5-10 minutes for the 16 subjects of the EYEDIAP database, using 50 images per subject. This is acceptable, as it has to be done only during the training of person invariant models from a dataset.

Eye image alignment for a test subject: for a test subject, finding his/her alignment parameters θ with respect to an already aligned training dataset (step above) takes around 10s when using 1 to 5 sample images, but there is much room for improving our implementation. Importantly, note again that this has to be done only once per subject, and that the same parameters can be used for different sessions over time. Finally, once the alignment parameters have been estimated, computing $\mathbf{I}(f(\mathbf{u};\theta))$ for each frame during tracking is very fast as it only corresponds to warping a small image.

Gaze estimation

The feature extraction is implemented as described in Sec. 5.3. For SVR we used the scikit-learn software [Pedregosa et al., 2011]. The kNN approach is based on a brute force search, but could clearly be improved, e.g., using a KD-Tree. The ALR and CALR methods implementation used the CVXOPT software to solve Eq. 5.4 and Eq. 5.5.

System speed

Overall, the gaze tracking takes around 100ms per frame. Note however this is a research implementation, where the most time consuming elements are the data pre-processing (RGB-D 3D mesh creation) and the head pose tracking. As mentioned in Section 3.5.1, the head pose tracking is CPU based and alone takes from 20 to 100ms (depending on the amount of head pose changes during consecutive frames).

The OpenGL based rectification takes 15ms. The gaze regression computation time depends on the used algorithm. For a particular case of using 1200 training samples (e.g., for an experiment from Sec. 5.6.1) the kNN method takes 25ms per eye, the H-SVR method takes 15ms per eye, whereas the R-SVR method takes 11ms per eye. The speed of ALR is heavily dependent on the size of the training set, e.g., when using 49 samples, the method takes 300ms, whereas it takes 3.5 seconds when using 150 samples. CALR suffers even further, as the optimization problem has twice the number of variables, e.g., when using 150 samples (per eye), the optimization takes 12 seconds. Remember that, in all cases (ALR and CALR), we solved the problem 7 times with different reconstruction bounds (see Section 5.3.3). Therefore, the reported time includes the 7 separate executions.

5.5.2 Gaze estimation dataset

We used the EYEDIAP database for all experiments on gaze estimation. This database has been described in full detail in Chapter 4. In this section we will here recall the main points

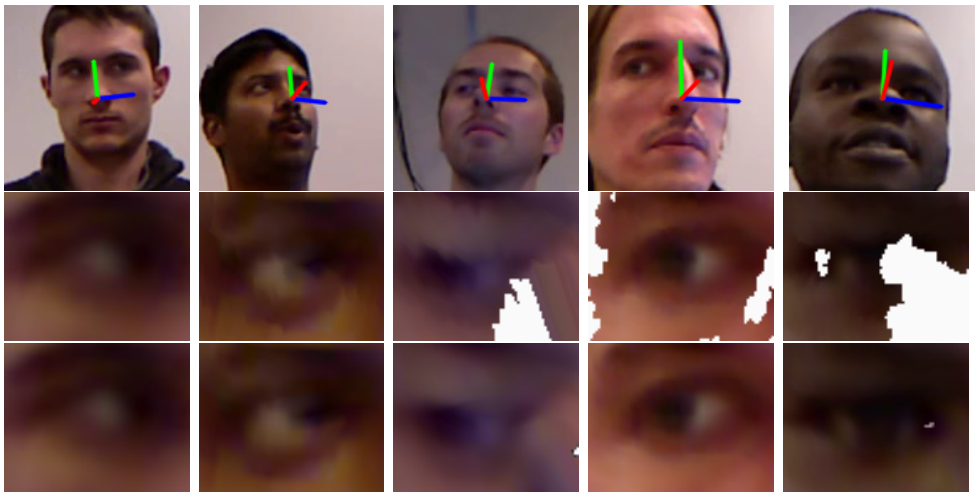


Figure 5.8: EYEDIAP samples and pose-rectified images. Each column correspond to one data sample. Per column we show i) the head pose, showing a region of 140 × 140 pixels from the RGB frame; ii) $\mathbf{I}^R$ images generated from the depth data (*DDM*); iii) $\mathbf{I}^R$ images generated from the template data (*TDM*). The first three columns correspond to sessions involving the *FT* target, whereas the last two correspond to samples for the *CS* target.

relevant to the experiments we conducted.

The EYEDIAP database contains 94 recording sessions lasting between 2,5 and 3 minutes and characterized by a combination of the main variables which affect gaze estimation algorithms: different participants, ambient conditions, visual target, and head pose activity. We mainly used the VGA depth and RGB modalities (RGB-D data), recorded from the Microsoft's Kinect™.

Participants

The recordings involve 16 different people (12 men, 4 women), whose age range between 20 and 45 and with wide ethnical diversity. Samples of the data can be seen in Fig. 5.8.

Visual target conditions

The recordings involved two main scenarios characterized by their gazing tasks. People had to follow either a "floating target" in the 3D space -a ball attached to a string- (*FT* condition), or they would look at a target continuously moving on a screen (*CS* condition). The *FT* case is a highly challenging problem but is interesting as it is very representative of HRI and HHI scenarios. As people were seated at a distance of $\approx 1.2m$ from the sensor, the typical eye image size is $\approx 13 \times 9$ pixels, and the gaze space is as large as $\pm 45^\circ \times \pm 40^\circ$. The head pose variations follow a similarly large range.

In the *CS* case, the person sat closer (at $\approx 0.9m$), leading to a a typical eye image size of $\approx 19 \times 14$ pixels. However, as the screen is well confined (spatially), the range of gaze directions (in

the 3D space, without considering head pose variations) is smaller, i.e., $\approx \pm 15^\circ \times \pm 10^\circ$. Some example images, before and after the pose-rectification procedure, are shown in Fig. 5.8

Head pose activity

For each of the target situations, the head pose of a person was controlled for two conditions. In the Static Pose (*SP*) case, participants were asked to keep the head approximately fixed. In the Mobile Pose (*MP*) case, they were asked to move and rotate the head in all directions (while still looking at the target), resulting in large head variations in the recorded data. Note again, participants were not requested to maintain a neutral expression, and the data involves people speaking or smiling.

Combined with the low eye image resolution, this makes the EYEDIAP dataset more challenging than many databases discussed in the literature while corresponding to conditions frequently encountered in HRI and HHI.

5.5.3 Gaze estimation experimental protocol

Gaze ground-truth

The EYEDIAP data comes with gaze related information. More precisely, the 3DMM of each of the participants was fitted using the method described in Section 3.2, and for each session, the head pose was tracked using the algorithm described in Section 3.3, allowing to obtain the eyeball center $\mathbf{o}^{\mathbf{WCS}}$ of an eye. Similarly, as described in detail in Section 4.4, the point of regard $\mathbf{p}_{PoR}$ was extracted in the **WCS**, either by tracking the ball or by knowing the target in the 3D calibrated screen, and used to derive the ground-truth gaze unitary vector in **WCS** as $\mathbf{v}^{\mathbf{g}t} \propto \mathbf{p}_{PoR} - \mathbf{o}^{\mathbf{WCS}}$.

Gaze annotated frames

It is important to note that *not all* frames in a session are annotated with $\mathbf{v}^{\mathbf{g}t}$ as the EYEDIAP dataset consists of non stop video recordings. Furthermore, to ensure the data at hand is reliable, we filtered the session frames according to the criteria described in detail in Section 4.4.6. This means we discard frames where: i) the ground truth is not available; ii) a gaze shift occurs; iii) self occlusion of the eye region is expected due to extreme head poses; iv) extreme gaze directions; or v) the subject is distracted or blinking. Note that the criteria iii and iv apply to each eye separately, meaning that in a given frame one eye's ground truth can be considered as valid while the other is not. As a result of these validity checks, the average number of valid frames per session is around 2400.

Performance measure

As recommended in Section 4.4.7, at each time frame of the evaluation set, we used the gaze angular error, computed as follows:

$$\epsilon_g = \arccos\left(\mathbf{v}^{\mathbf{WCS}} \cdot \mathbf{v}^{\mathbf{g}t}\right) \tag{5.10}$$

where $\mathbf{v}^{\mathbf{WCS}}$ is the estimated gaze direction. Aggregated performance was obtained by computing the mean angular error over the test frames of each session. The average and standard deviations were then computed from the results obtained from the relevant sessions.

Missing measurements

When using depth measurement for the rectification (see Sec. 5.2.3), some pixels of the cropped image $\mathbf{I}^C$ may not be associated with any RGB measurement (see Fig. 5.8). This can be due to large head poses causing self occlusion, or missing depth data measurements in the eye region. To handle this situation, the gaze classification methods were updated as follows. In the kNN and ALR cases, the missing dimensions were simply excluded in the distance computation (kNN) or in the reconstruction (ALR[6]). In the R-SVR and H-SVR cases, the missing pixels were simply replaced by the average of the available measures.

Experimental protocol

In all the evaluations, the training data is disjoint from the test data. This was obtained either by training on one/several session(s), and testing on another one (e.g. for testing head pose or person invariance), or by splitting temporally a given session in two halves. The used sessions will be detailed per experiment.

When defining the evaluation set, we used the convex hull based filtering[7], which, as explained in Section 4.4.6, is required to avoid introducing noise into the evaluation of appearance based methods. This was further motivated by the fact that, in the head pose invariance experiments with the screen (*CS* case, Sec. 5.6.2), there was a mismatch between the training and test gaze angles range at times during the sessions. As the screen is a small object (within the larger 3D space), the training samples collected using a static head pose only cover a small gaze space region, whereas sessions with head pose variations induce a larger coverage as the screen region moves within this space following head movements, causing the aforementioned mismatch. For head pose invariance experiments in the *CS* case, this filtering discarded ≈40% of the test samples. Nevertheless, the remaining samples are still diverse in terms of combined head pose and gaze directions. Note that (i) in the other experiments (*FT* target, person invariance), excluded frames represented less than 5% of the test frames and (ii), in all cases, as the training and test samples are the same across different gaze regressions methods, results

[6] As in ALR a dimension is obtained by averaging over pixels within a region, the dimension was discarded if the proportion of availble pixel values was less than 30%

[7] Note these evaluations are done in terms of gaze angles (elevation and yaw) defined with respect to the **HCS**

Table 5.1: Summary of results on mean angular gaze error (°) for the floating target conditions (*FT*). For a given experimental protocol we report, per evaluated method, the mean (top) and standard deviation (bottom) computed over all relevant sessions for the given conditions: i) *SP-PS*: static pose with person-specific gaze models (Sec. 5.6.1) ii) *MP-PS*: mobile pose with person-specific gaze models (Sec. 5.6.2) iii) *SP-PI*: static pose with person invariant model (Sec. 5.6.3) iv) *MP-PI*: mobile pose with person invariant model (Sec. 5.6.4). Acronyms: *SP* (static pose) - *MP* (mobile pose). *PS* (person specific) - *PI* (person invariant). D (*DDM* data-driven rectification) - T (*TDM* template-driven rectification). *NA* (no alignment) - *FL* (automatic eye corners detection based alignment) - *EC* (manual eye corners annotation based alignment) - *A* (SDIPA-based supervised alignment) - *A5* (SDIPA-based supervised alignment using only 5 samples for the test subjects).

	SP-PS	*MP-PS* pose invariance		*SP-PI* person invariance					*MP-PI* pose and person invariance				
Method	-	D	T	*NA*	*FL*	*EC*	*A*	*A5*	*NA*	*FL*	*EC*	*A*	*A5*
HP	28.6	23.0	23.0	28.0	28.0	28.0	28.0	28.0	23.6	23.6	23.6	23.6	23.6
	3.0	4.0	4.0	2.7	2.7	2.7	2.7	2.7	3.9	3.9	3.9	3.9	3.9
kNN	6.4	9.8	9.8	12.2	11.9	10.9	10.0	**10.0**	13.7	13.4	13.3	12.2	12.4
	1.5	2.6	2.7	3.2	3.5	2.7	2.1	1.8	3.2	4.0	3.5	2.7	2.5
ALR	6.2	11.5	10.3	13.7	-	-	-	-	-	-	-	-	-
	1.9	2.2	2.0	4.6	-	-	-	-	-	-	-	-	-
H-SVR	6.0	9.3	9.0	11.8	**11.3**	10.7	**9.8**	10.4	13.0	12.6	12.0	11.6	11.8
	1.9	2.2	2.1	3.8	3.2	2.9	3.1	3.5	2.5	3.1	2.3	2.2	2.5
R-SVR	**5.6**	**8.5**	**9.0**	**11.6**	11.6	**10.6**	10.5	11.0	**11.7**	**12.4**	**12.0**	**11.4**	**11.6**
	1.7	1.8	2.7	5.1	3.5	3.4	3.6	3.7	2.7	3.2	2.6	2.4	2.4
CALR	5.9	11.4	10.4	-	-	-	-	-	-	-	-	-	-
	1.6	2.1	2.4	-	-	-	-	-	-	-	-	-	-

accross methods are directly comparable. Protocol elements specific to a given experiment are presented in the result Section.

5.6 Results

In this section we describe the results obtained using our framework. Sections 5.6.1 to 5.6.4 presents the results of the gaze estimation experiments conducted on the EYEDIAP database, discussing the different aspects (appearance models, pose and person invariance, alignment, etc.) of our methodology. These results are summarized in Tables 5.1 and 5.2.

5.6.1 Static pose and person specific conditions (*SP-PS*)

In this section we compare the regression methods assuming the model is trained and tested for the same person and under minimal head pose variations. There are 19 sessions for the *FT* target, 14 sessions for the *CS* target. In each session, the algorithm was trained using the first -temporal- half, while the evaluation was done in the second half. This means that, on average,

Table 5.2: Summary of results on mean angular gaze error (°) for the screen target conditions (*CS*). We report the mean (top) and standard deviations (bottom) computed over all sessions relevant for a given condition. For acronyms, see Table 5.1.

	SP-PS	*MP-PS* pose invariance	*SP-PI* person invariance					*MP-PI* pose and person invariance				
Method	-	T	*NA*	*FL*	*EC*	*A*	*A5*	*NA*	*FL*	*EC*	*A*	*A5*
HP	13.5	15.7	13.0	13.0	13.0	13.0	13.0	15.7	15.7	15.7	15.7	15.7
	2.9	4.2	2.5	2.5	2.5	2.5	2.5	4.2	4.2	4.2	4.2	4.2
kNN	2.9	4.2	8.7	7.8	7.6	6.6	6.6	9.8	9.0	8.6	7.6	7.8
	1.2	1.3	3.5	2.2	2.8	2.9	3.2	2.6	2.0	2.5	2.3	2.7
ALR	2.4	4.8	9.2	-	-	-	-	-	-	-	-	-
	0.9	1.7	3.9	-	-	-	-	-	-	-	-	-
H-SVR	1.9	**3.5**	**5.8**	**6.2**	**5.7**	**4.9**	**5.1**	**6.8**	**6.8**	**7.0**	**5.7**	**6.0**
	0.8	1.3	3.0	2.5	2.7	2.2	2.1	2.6	1.9	2.2	1.9	2.1
R-SVR	**1.7**	3.6	6.6	6.4	6.6	6.0	6.6	7.6	7.1	7.3	6.4	6.9
	0.8	1.4	3.1	2.7	3.6	2.6	3.5	3.3	3.0	3.2	2.4	3.3
CALR	2.4	4.7	-	-	-	-	-	-	-	-	-	-
	0.9	1.7	-	-	-	-	-	-	-	-	-	-

around 1200 samples are used for training[8] and around 1200 are used for testing. Note that this corresponds to Benchmark 1 from the EYEDIAP database (cf. 4.5.1).

Gaze accuracy

The first column in Tables 5.1 and 5.2 show the mean angular errors averaged for the relevant recording sessions using the *FT* or *CS* conditions respectively. In addition, Fig. 5.9 provides the recall-error curve obtained for each method for the *FT* condition.

We can first notice from the results obtained using only the head pose (HP) as gaze approximation that there are large gaze variations within the data. This variability is much larger in the *FT* case, where the target was moved in the 3D space region in front of the subjects, than in the *CS* screen gazing situation. Thus, although the gaze estimation methodology is the same in both cases, the error of the different methods is significantly lower in the *CS* (1.7 to 3°) than in the *FT* case (around 6°). This difference highlights that the choice of the task and data has a large impact on the performance, and that in general errors can not directly be compared in absolute terms without taking into account the sensor and experimental conditions. Nevertheless, as shown by Fig. 5.9, more than 85% of the gaze errors are below 10^o in the difficult *FT* conditions.

When comparing the different regression methods, we can notice that the SVR methods perform better than kNN, ALR and CALR, and that, under the current experimental conditions

[8] Note that for ALR and CALR, the number of training samples was limited to 150. Otherwise the test time is prohibitively large (see Section 5.5.1).

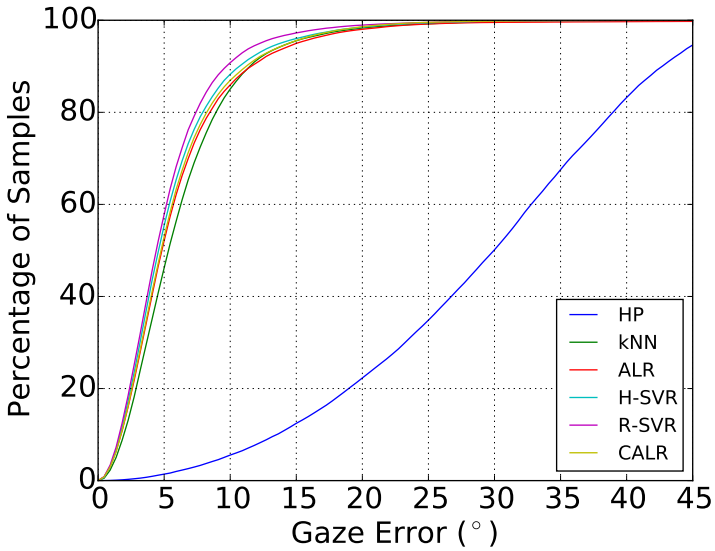


Figure 5.9: Recall-error curve obtained for each of the gaze estimation methods. *FT* target, person-specific model (*PS*), minimal head pose variations (*SP*).

(*SP-PS*) the retinex features show better performance than the mHoG. Note also that although there are variabilities amongst the sessions, with for instance results ranging from 1.1° to 3.2° in the *CS* case and 3.1° to 9.9° in the *FT* case using R-SVR, this R-SVR method is obtaining the best results in 15 over 19 sessions in the *FT* case, and 13 out of 14 in the *CS* case.

Gaze error distributions

Fig. 5.10 displays the estimation error distributed according to the ground truth gaze angles. HP method is not shown, but its error is equal to the absolute value of the ground truth. The plots show that errors are well distributed over the large range of gaze values. Interestingly, we can note that kNN has a flatter error distribution w.r.t. the ground truth. In particular, it has the lowest errors at large angles, followed by R-SVR. The ALR and CALR methods, on the other hand, degrade faster for the largest angles. CALR still gives better results than ALR, but it starts to degrade further at large pitch angles. Overall, these plots validates an important advantage of appearance based methods in general, as these are capable of gaze estimation even when the iris is heavily occluded by the eyelid (e.g., when the person is gazing down), which is not the case of geometric based methods relying on feature tracking.

Number of training samples

We also evaluated the gaze estimation error as a function of the amount of training data. In these experiments, the training set was regularly sampled (in time) to obtain the desired number of training samples. Note that the training samples are the same for all methods, and that the test data remained the same as in previous experiments.

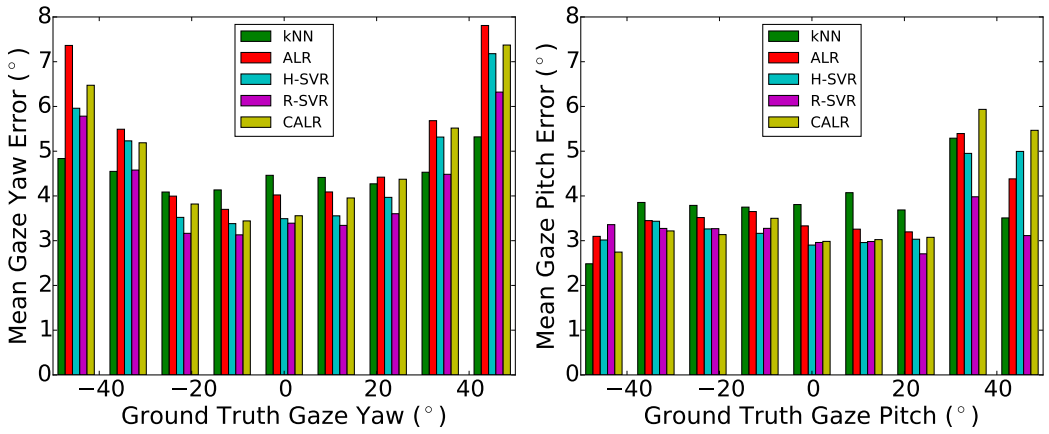


Figure 5.10: Gaze error distribution in function of the ground truth gaze angles. Conditions: *FT* target, person-specific model (*PS*), minimal head pose variations (*SP*).

The results are shown in Fig. 5.11. Even though R-SVR is outperforming the other methods when more training data is available, ALR and CALR showed to be advantageous when using a smaller training set (less than 50 samples), with CALR slightly better than ALR. However, these methods do not scale well for larger amounts of training data, due to the computational complexity of solving the constrained L1 minimization. Therefore, this experiment suggests the ALR and CALR approaches are adequate whenever a short calibration session is possible.

5.6.2 Head pose invariance (*MP-PS*)

In this section we present experiments related to the head pose invariance capabilities of our framework. The protocol we followed corresponds to Benchmark 2 of EYEDIAP (see Section 4.5.2). Note that for each of the 19 (*FT*) or 14 sessions (*CS*) used in Section 5.6.1, there is an equivalent recording session (same person and visual target) involving head pose variations rather than a static pose. Therefore, we used as training set the session involving a static head pose and as evaluation set the equivalent session with head pose variations, each of them comprising 2400 valid samples on average. Note also that this procedure can only lead to correct results if the proposed methodology is indeed head pose invariant thanks to the generation of pose-rectified eye images.

Two rectification procedure are compared in the *FT* case: the one relying on the sensor depth data (*DDM*, D), and the one relying on the fitted template mesh (*TDM*, T), as described in Section 5.2.3. For the *CS* case, only the template based one could be applied. In this situation the participants were at a closer distance to the sensor, near its sensing limit, and there were too often missing depth values in the eye region (see examples in the middle row and right of Fig. 5.8)[9].

[9]Thus, all reported results for *CS* are with the template rectification. For the *FT*, the default rectification was with the data driven rectification *DDM*. Note that in the static pose *SP*, differences between the two rectification

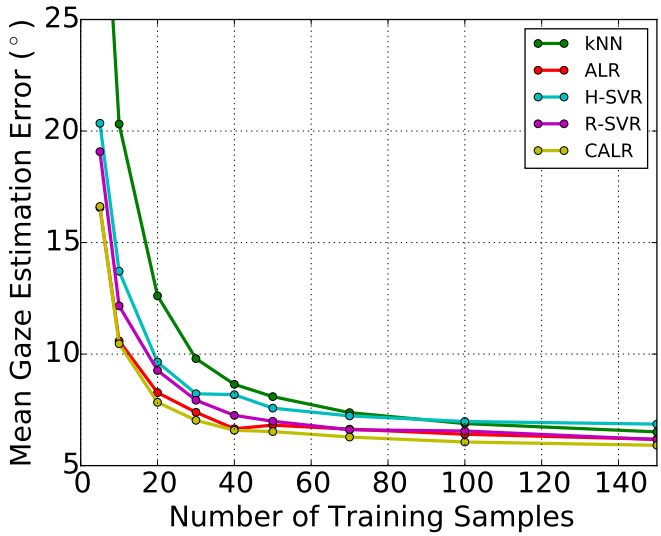


Figure 5.11: Mean angular error vs. number of training samples. Conditions: *FT* target, person-specific model (*PS*), minimal head pose variations (*SP*).

Results

They are shown under the "*MP- PS*" columns in Tables 5.1 and 5.2. Notice first that the HP baseline presents slightly lower error in comparison to the *SP* case (23° vs 28.6°). This is because participants tended to reduce large gaze variations when head movements were possible, hence the gaze and head pose are more correlated for these sessions. Nevertheless, the high error for HP indicates that there is still a wide range of gaze variations.

The results show a degradation of the results as compared to the static case (+3° in *FT* condition, around +1.8° in the *CS* case). This is very reasonable, considering that more than 50% of the samples have a head pose larger than 20°.

Again, the two SVR methods perform better. The ALR and CALR methods, whose performance is very close, seems to suffer more than the other approaches from the head pose changes. This is particularly true in the data driven case, and might be due to the loss of dimensions in the eye representation when no depth measurements are available in some eye regions. This is confirmed by comparing the error distributions according to the head pose, shown in Fig. 5.12: ALR and CALR errors are higher in the *DDM* case than in the *TDM* for a head yaw angle near to -10 or -20°. Also, CALR degrades more than ALR at larger head yaw angles, possibly as, between the two eyes, the occluded eye might influence negatively on the visible eye. Notice that at head yaw angles further than -20° the right eye gets more and more occluded by the nose, whereas at positive and larger angles (up to ≈ 50°) the right eye remains visible. These error distributions also show that, as expected, the errors increase in terms of the head pose angle. For our methodology, the source of errors are diverse: missing depth values, rendering

methods are nearly indistinguishable, as almost no 3D rotation is applied.

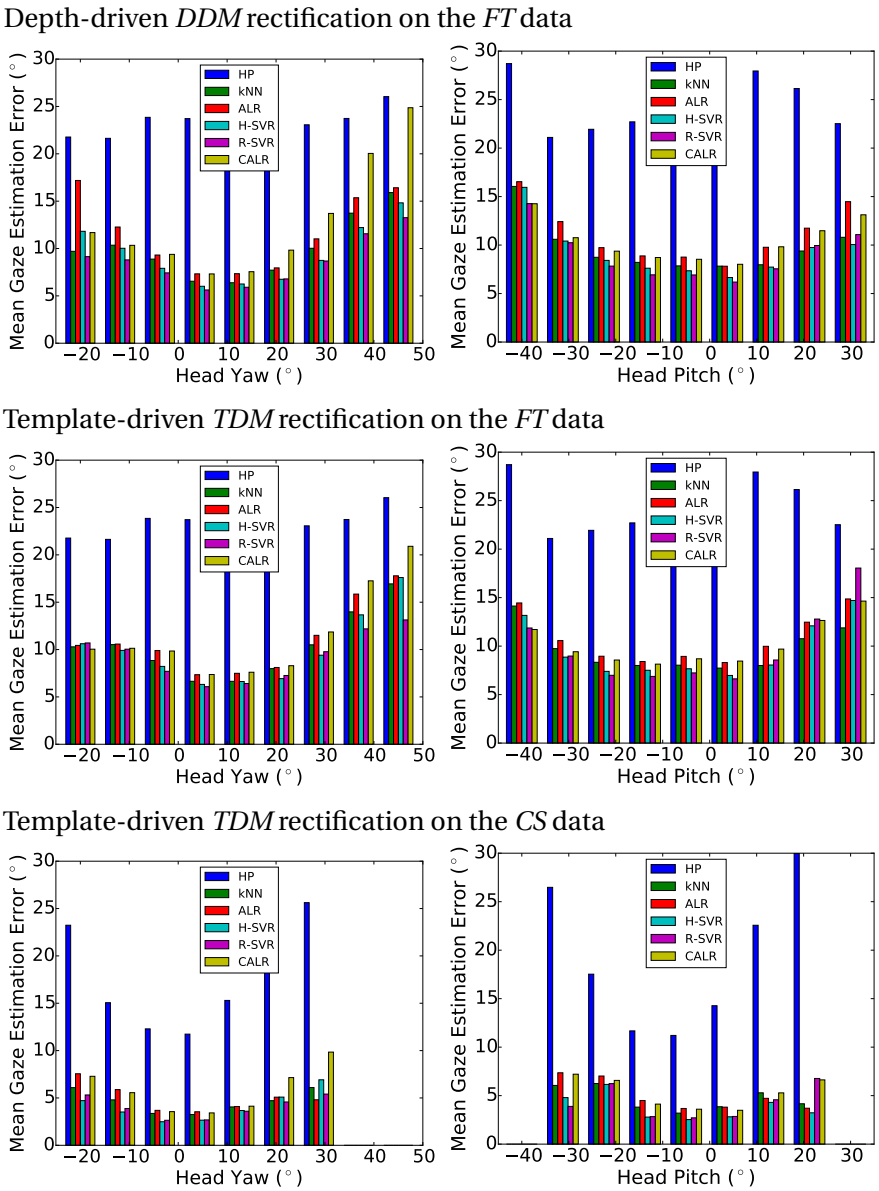


Figure 5.12: Gaze error distribution for the right eye in function of the head pose for the *FT* and *CS* cases and different pose-rectification rendering methods.

artifacts due to depth-noise, and self-occlusions.

Finally note that, although the two rectification methods perform in par overall[10], the template method seems to suffer more from larger head pitch (errors near +20° and +30°) when looking up.

[10] For instance, in the R-SVR case, despite a gain of 0.5 for the depth driven approach, it only performs better than the template based method in 10 sessions out of 19.

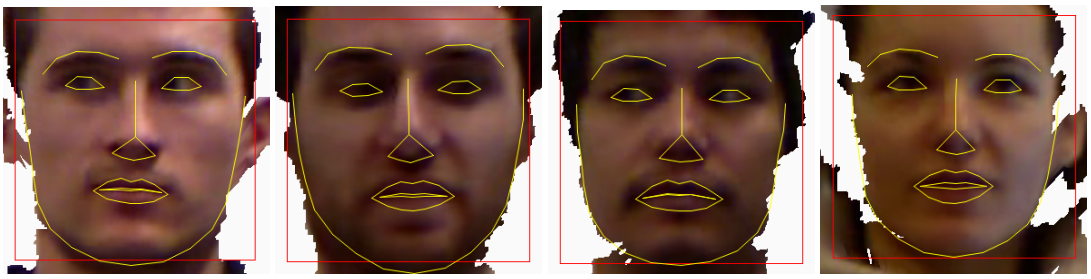


Figure 5.13: Automatic landmark detection on the pose rectified face image using the method by Kazemi and Sullivan [2014].

5.6.3 Static head pose, person-invariance (*SP-PI*)

To evaluate the person invariance case, we conducted a leave-one-person-out cross-validation on the sessions involving minimal head pose variations (*SP* data), which corresponds to the EYEDIAP's Benchmark 3 (cf. Section 4.5.3). This means that in each of the N experiments (where N is the number of sessions for the *FT* or *CS* case), there are around $2400 \times (N-1)$ samples available for training[11] and around 2400 for testing.

The results for *FT* and *CS* are reported under the *SP-PI* columns from Table 5.1 and 5.2 respectively, and differ on which alignment strategy was used, if any. The obtained results can be compared to the person-specific case on the same data (*SP-PS*)[12].

Overall results

As can be noticed, there is an important error loss (around 5.5° for *FT*, 4° for *CS*). Overall, the error is larger than when considering head pose variations, suggesting that with low resolution images, eye appearance variability due to different users are more important than those due to head pose changes (even large) after our proposed rectification. The errors are not distributed equally across subjects: there are difficult cases for which the errors rise to 15.7, 17.1 and 26.4 degrees for R-SVR/*FT*, as it can be observed by its larger variance of 5.1° (*NA* case). Note that, in particular, ALR performs poorly and the mandatory selection process described in Sec. 5.4 (here obtained from 1 out of 50 samples) was prohibitively slow. For these reasons, we did not evaluate the alignment techniques with ALR. The process is even slower for CALR, thus we did not evaluate this method for person invariant gaze estimation experiments.

Alignment methods comparisons

We evaluate four types of alignment: "*FL*" correspond to an alignment based on an automatic facial landmarks detection algorithm, "*EC*" correspond to an alignment based on manually

[11] For the SVR methods we limited the training set to 1200 samples as using the full set was prohibitively slow.

[12] With the slight difference that the evaluation is conducted on all samples of the test subject's session, instead of only the second half in the person-specific case.

annotated eye corners, "*A*" is our proposed synchronized delaunay implicit parametric alignment whereas "*A5*" is the same approach but using only 5 samples for the alignment of the *test* subject. *NA* correspond to no alignment (more details in the following).

The first tested method is *FL*. In this case we applied the facial landmarks detection method of Kazemi and Sullivan [2014] on the pose-rectified facial images, as shown in Fig. 5.13; although in practice this method showed good stability on this type of images, we obtained the eye corners position for over 100 frames and computed their average to account for minor variations. Considering future improvements on automatic landmarks localization algorithms, we also evaluated the *EC* case, which means that 10 to 15 eye image samples were annotated *manually* with the eye corners[13]. In both cases this was used to register the eye images in a canonical view from the average eye corners position.

Overall, the *FL* strategy brings minor improvements to the *FT* scenario; although it has a similar behavior in the *CS* case, it actually degrades the accuracy for the H-SVR method. The gain is nevertheless larger for the *EC* strategy, with a gain of 1° in *FT*, but surprisingly almost no gain in *CS*, except for the kNN method.

Alternatively, the proposed Synchronized Delaunay Implicit Parametric Alignment (denoted *A* and *A5* in Tables 5.1 and 5.2) can be applied. In *A*, all the test gaze samples were used for alignment (including the test subject), so the method performance can be somehow considered as an oracle. In the *A5* case, 5 samples whose gaze values were close to 0 were used to align the eye of the test subject with an already aligned dataset.

Notice that the *A* vs. *A5* comparison is motivated by possible applications. "*A*" can be interpreted as the ideal case which would lead to the best alignment under this approach. "*A5*", on the other hand, is representative of a more practical scenario, where it might not be possible to collect enough gaze annotated samples to train a gaze estimation model specific to the test subject, but it might still be possible to collect a small set of gaze annotated samples. Notice that such possibility may not require any cooperation from the test subject. In Chapter 7 we study an application which exhibit these characteristics.

The results shown in Tables 5.1 and 5.2 demonstrate the interest of the proposed alignment method: *A* improves the result in both the *FT* and *CS* case, even outperforming the *EC* manual alignment case. Fig. 5.14 illustrates qualitatively the alignment effect. Despite significant differences in eye appearances, the eye alignment is visually better after *A* than before.

Finally, results of the *A5* case show that the use of a minimal set of labeled samples can bring a good gain in the result: the best performing technique in *FT* (kNN) undergoes a reduction of 2.2° as compared with no alignment (*NA*), improving the results in 18 out of 19 sessions; in *CS*, the gain is of 0.7° (for H-SVR), improving the results for 9 out of 14 sessions.

[13] Doing such annotation was not so easy in practice. Given our image resolution, determining visually the location of an eye corner is difficult, thus the need for multiple annotations.

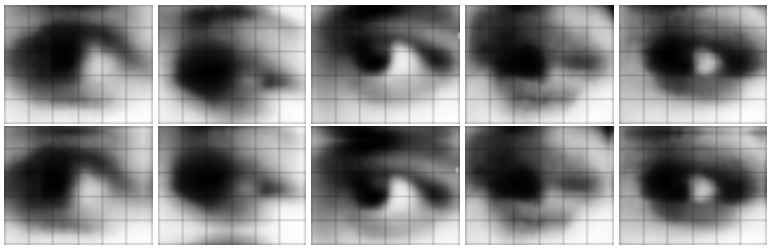


Figure 5.14: Alignment example. All samples share the same gaze direction **y**. The images are shown before (top row) and after (bottom row) alignment on the dataset, for different subjects (one per column). Note for instance the discrepancy in the height location of the iris before alignment (too low in the 2nd sample, too high in the 3rd).

5.6.4 Pose variations and person invariance (*MP-PI*)

Finally, we evaluated the performance of our approach in the most general case: data with head pose variations (*MP* sessions), and using a person invariant gaze estimation model. In this case, for a test subject, the data from the static pose *SP* of all other subjects were used as training data. The size of the training and test sets are the same than in the *SP-PI* case. Similarly, the alignment parameters employed were those estimated in the static case (cf previous subsection).

The results are reported in Tables 5.1 and 5.2 under the "*MP-PI*" columns, and can be compared to those reported when using a person specific model (*MP-PS* condition). Looking at the best technique for *FT* (R-SVR) and *CS* (H-SVR), the following comments can be made. The person invariance situation increases the errors in a similar fashion than in the *SP* case: +3.2° in *FT*, +4° in *CS*. The proposed alignment approaches *A* (resp. *A5*) contribute to reduce the error: −0.3° (*A5*, −0.1°) in *FT* for the best performing method (note that the decrease is larger with the other methods, around −1.2° for H-SVR or kNN in the *A5* case); −1.1° (*A5*, −0.7°) in the *CS* case.

5.7 Discussion and future work

We have proposed methods to address the head pose and person invariance problems and have validated them through extensive experiments. In this section, we present and discuss the limitations of the the proposed methodology and how it could be extended and improved in diverse ways in future work.

An exhaustive comparison in terms of features and regression algorithms has not been conducted in this paper, as our purpose was but to validate our contributions using the best and representative features and algorithms found in the literature, as motivated in Section 5.4. This leaves room for future studies evaluating whether in our framework and scenarios, other types of features such as local binary patterns, the possible exploitation of color information, or combination of features (as done by Schneider et al. [2014] for instance), could improve the

results. In this direction, it could also be relevant to evaluate whether there is an impact of the pose rectified eye image size on the performance error, taking into account the distance at which the system is expected to operate; or similarly, evaluate the impact on accuracy of the amount of training data, e.g., by using less than 15 people, or by collecting more data to see at which level the method saturates. Such studies could be facilitated and compared to our work thanks to the use of our using publicly available database.

The alignment, which was implicitly defined within a person's head frame, was intended to correct eye image cropping inconsistencies across subjects, when building a person invariant gaze model. A benefit of the method was that it can compensate for 3DMM fitting semantic errors across subjects, even if the the eye corners are not well located or difficult to locate given the image resolution. Importantly, by exploiting the gaze input to conduct the alignment, the methods implicitly strive to align the actual eyeball positions across subjects, which is what the gaze alignment step should aim for[14]. Notice, in the experiments in this chapter we used the offline 3DMM fitting procedure, mostly to keep consistency with what is provided to users of the EYEDIAP database. However, it is expected that such semantic fitting errors may be more common and could be well corrected when using the online fitting method described in Section 3.4, and that the proposed alignment would be even more beneficial in such cases.

In this work, a single alignment was performed per subject. However, in practice, we do observe as well frame to frame misalignment errors coming from small ICP fitting differences across frames, due to missing or noisier depth information (e.g., at larger depth), face deformation in the eye region, or erroneous pose estimation. To handle this, it would be interesting in future work to explore frame by frame alignment methods, e.g., through eye image stabilization leveraging on robust optical flow estimation, image registration, and automatic landmark detection methods.

Finally, note that, even though the alignment function f we used here is a translation, we hypothesize that other transforms may consider more geometric variability. In particular, including a scale may model eye/eyeball size variations.

The eye image pose rectification plays a role as well in our approach, especially when the eye goes towards more profile views. The *TDM* template method would profit from a 3DMM model with a tighter fit in the eye region. Local non-rigid registration methods, or unsupervised frame matching and averaging could be used there. As the depth noise level makes this challenging, RGB information could be exploited as well. Also, depth information could help handling self occlusion by the nose. The *DDM* depth driven method could alternatively make use of depth filling methods and depth smoothing, to maximize the region with texture information in the pose rectified image, and to reduce artifacts (see Fig. 5.8). Note the *TDM* approach implicitly has this function.

Finally, we want to emphasize that our proposed approach could be exploited in diverse

[14] and is different than aligning the eye corner features

manners for many applications. It could be used with no cooperation from the user whatsoever, meaning the overall system and person invariant gaze models are used as is, for a new test subject. Alternatively, a minimal cooperation protocol could be defined to obtain the needed alignment data, either explicitly, e.g. requesting the participant to fixate at the camera for a few seconds [Oertel et al., 2014], or implicitly through an agent (e.g., a robot) persuading the subject to do such actions either by a direct request or by leveraging on gaze priors on non verbal human behaviors in a dialogue situation. In another direction, a third person could annotate higher level gaze semantics (people gazing at known targets). An example of this methodology will be shown in Chapter 7.

5.8 Conclusions

In this chapter we have proposed a framework for the automatic estimation of gaze in a 3D environment. We address two of the main factors which directly influence the eye image appearance, and which lead to a decrease of the gaze estimation accuracy: i) head pose variations and; ii) inter-user appearance variations.

For the challenge of head pose variations, we have proposed a framework which rectifies the captured eye images into a canonical viewpoint. To this end, we rely on depth information to accurately track the 3D head pose. Given an accurate head pose, we proposed, and evaluated two strategies for the viewpoint correction: either based on the depth measurements or the fitted 3D facial mesh.

To address person invariance, we have conducted extensive experiments evaluating state-of-the-art appearance based gaze estimation algorithms within our framework under several conditions. We have also addressed the problem of eye image alignment as it has a direct link with the person invariance problem. We therefore proposed a new method for the inter-subject eye image alignment from gaze synchronized samples, and we validated its advantage with respect to other strategies.

We believe the proposed solution is highly valuable in many types of scenarios in human human interaction or in human robot interaction applications.

6 Geometric Generative Gaze Estimation (G^3E)

6.1 Introduction

In the literature review presented in Chapter 2, we discussed the many methods proposed in the past to address the gaze estimation problem. Two main methodologies were identified: appearance based methods (ABM) and geometric based methods (GBM). Each of these two methodologies have their own advantages and disadvantages.

Appearance based methods for gaze estimation are of interest as, by modelling a direct regression from the eye image appearance into the gaze parameters, they circumvent the extraction of local eye features. Therefore, these methods are more suitable for the low resolution sensing conditions which are to be expected in less controlled scenarios in HHI, HRI and HCI.

However, as discussed previously, these methods often require either -per session- training data, resulting in overfitting to the person and conditions used during the training phase, or require large sets of training data to learn a generic model which is intended to be invariant to the many elements which have an impact on the resulting eye image appearance (cf. Fig. 1.3). However, ABM still generalize poorly, as their invariance properties are limited to the conditions found in the training set.

In Chapter 5 we proposed strategies to improve the generalization of ABM, in particular, to address variations due to head pose and user specific appearance. This led to the creation of generic gaze estimation models which are of high value in non-cooperative scenarios. However, even though a generic model may indeed attain certain degree of generalization,it is not trivial to adapt such model to the test data to further improve the gaze estimation accuracy, in particular, to the given user and sensing conditions (illumination, resolution, contrast).

Geometric based methods do not suffer from this limitation as they rely on parametric eye geometric models dedicated to the gaze estimation problem. Their "adaptation" phase is intended to infer the subject specific geometric parameters (calibration). However, the success of GBM is strongly dependent on the capability to reliably and accurately extract local eye features, e.g., corneal reflections, or the iris/pupil center, prior to the model fitting step or the

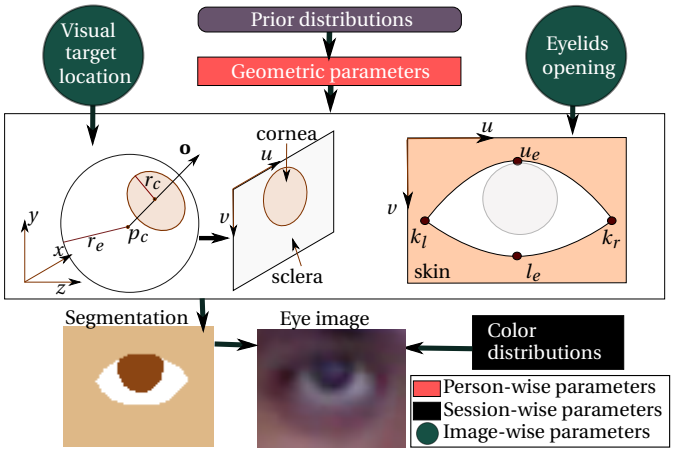


Figure 6.1: Geometric generative gaze estimation overview. See text below.

estimation of gaze. For this reason, these approaches are restricted to high quality sensing conditions, such as infrared setups and/or high resolution sensors.

In this Chapter we propose a new gaze estimation paradigm, which we call *geometric generative gaze estimation* (G^{3}E). This approach relies on a probabilistic generative process to model the appearance of eye images built over a geometric parametric eye model.

The G^{3}E model is illustrated in Figure 6.1. The generative process links the action of gazing at a visual target and the eyelids movement, which are *image specific* quantities, with a semantic segmentation of the eye region. The generated segmentation image is furthermore driven by the *person specific* eye geometry. Then, the segmentation image, combined with class/region specific color distributions, leads to the generation of colored eye images. Notice that the color distributions can be seen as *session specific*, as they depend on the ambient conditions.

This generative process can then be used to evaluate the validity of a given geometric configuration (including gaze) according to likelihood measurements on the eye image, and possibly, visual target location observation(s). Notice that this generative process depends on, but more importantly, decouples, the parameters describing the user specific eye and eyelids geometry, the image-wise parameters related to the gaze action and eyelids opening, and the ambient/session specific parameters, modelled by the color distributions.

Therefore, the principle of G^{3}E differs significantly from the appearance based and geometric based paradigms. Nevertheless, it has the best characteristics of both ABM and GBM, while overcoming their most important limitations, as explained below:

- **Low-resolution sensing**. As G^{3}E relies on evaluations conducted over the entire eye image (as in ABM), through likelihood evaluations based on the semantic parametric segmentation, this model circumvents the local eye features tracking problem. Therefore, this makes the method suitable for low resolution sensing, in contrast to standard GBM.
- **Gaze extrapolation**. This approach builds over an explicit eye geometric model. Therefore, it is capable of extrapolating to gaze directions which were not seen in the training

phase. Notice that, in contrast, this is an important limitation of ABM.

- **Training from fewer samples**. G^{3}E incorporates prior distributions on the geometric parameters (see Fig. 6.1). This allows to regularize on the plausible eye geometric solutions, making the model to adapt quickly to the given user from fewer observations. Therefore, this results on shorter calibration sessions, in comparison to ABM.
- **Adaptation**. This model effectively decouples the person-wise, session-wise and image-wise parameters. This allows to design specific calibration sessions to retrieve the person specific geometry. Furthermore, the decoupling allows to potentially adapt the model to different sensing and ambient conditions, by replacing/adapting the color distributions. This enables the gaze sensing accross multiple sessions, without the need to re-estimate the person specific geometric parameters.

Furthermore we here developed the G^{3}E model in the context of the head pose invariant gaze estimation framework, developed in Chapter 5, which relies on remote RGB-D consumer sensors. Therefore, the overall approach is head pose invariant and it is valuable for gaze reasoning in a 3D environment, a desired property in HHI, HRI and HCI applications.

The rest of this Chapter is structured as follows: the head pose invariant framework is summarized in Section 6.2. The G^{3}E model is detailed in Section 6.3, followed by the inference scheme in Section 6.4. Section 6.5 presents the experiments we conducted to validate the method and its advantages. Finally, in Section 6.6 we conclude this Chapter. Further details can be found in Appendix B.

6.2 Gaze estimation from RGB-D sensors

In this Chapter, the G^{3}E model builds on top of the head pose invariant gaze estimation framework based on RGB-D sensors, described in Chapter 5. Here, we briefly summarize the processing steps relevant to G^{3}E. We again assume a 3D user specific face mesh (template) is built offline by fitting a 3D Morphable model (3DMM) to multiple RGB-D data samples. The approximate eyeball center $\mathbf{o}^{\mathbf{h}}$, defined with respect to the head coordinate system (**HCS**), is obtained from the 3DMM topology. Then, in an online stage, the following steps are executed:

1. The 3D head pose $\mathbf{p}_t$ is obtained by registering, frame-by-frame, the face template to depth data using the ICP algorithm, resulting, for frame t, in the 3D head rotation and translation $\mathbf{p}_t = \{\mathbf{R}_t, \mathbf{t}_t\}$, referred to the world coordinate system (**WCS**).
2. Assuming a calibrated RGB-D setup, the RGB-D frame is transformed to a textured 3D mesh. We then re-render the texture, lying on the 3D data surface, using the inverse head pose parameters $\mathbf{p}_t^{-1} = \{\mathbf{R}_t^\top, -\mathbf{R}_t^\top \mathbf{t}_t\}$. This results in facial images as if the head was static and in front of the camera. From the position of $\mathbf{o}^{\mathbf{h}}$ we crop an eye image from the frontal looking facial texture, resulting in pose-rectified eye images.
3. The gaze direction is estimated from the pose-rectified eye images using the proposed G^{3}E model, which will be described in Section 6.3.
4. The gaze direction is transformed back to the **WCS**, according to the head pose.

Table 6.1: List of symbols related to the G^{3}E model

Symbol	Description
$\mathbf{I}; (u, v)$	Image *index* and pixel *coordinates*
p_c	Eyeball rotation center
$\kappa = [\phi_\kappa, \theta_\kappa]^\top$	Visual axis deviation
d	Nodal point distance from p_c
$\mathbf{a} := \{\kappa, d\}$	*Axial* parameters
r_e, r_c	Eyeball and cornea radii
$k_l = \{k_{lu}, k_{lv}\}$	Left eye corner in image coordinates
$k_r = \{k_{ru}, k_{rv}\}$	Right eye corner in image coordinates
$k_{lr} = \{k_l, k_r\}$	Left and right eye corners
$\mathbf{s} := \{r_e, r_c, k_l, k_r\}$	*Structure* parameters
$\mathscr{U} := \{\mathbf{s}, \mathbf{a}, p_c\}$	User specific geometric parameters
p	Visual target 3D position
$\mathbf{o} = [\phi, \theta]^\top$	Optical axis orientation
u_e, l_e	Upper and lower eyelid opening
$\mathbf{m} := \{\mathbf{o}, u_e, l_e\}$	*Movement* parameters
Λ_l	Color distribution parameters for class l
c	Observed color at pixel u, v
$\lambda \in \{0, 1\}$	Occlusion state for pixel u, v

6.3 Geometric generative gaze model

In this section the proposed model is described. First, an overview of the method is discussed, followed by details on the eye geometric model and parametric segmentation function. The generative framework is then described followed by the inference strategy.

6.3.1 Model overview

The proposed approach is summarized as a block diagram in Figure 6.1. Before describing the method, notice that all measures can be referred to the head coordinate system (**HCS**) due to the pose-rectification procedure described in Section 6.2. This makes it possible to deal with head fixed quantities. Furthermore, as a consequence of relying on depth data, there is no scale ambiguity in the pose-rectified eye images, and the pixel size, in meters, is known.

The model is characterized by user specific parameters $\mathscr{U} = \{p_c, r_e, r_c, \kappa, d, k_{lr}\}$, which define the fixed eye geometry (all notations are defined in Table 6.1), and image specific parameters $\mathbf{m} = \{\mathbf{o}, u_e, l_e\}$ related to the time changing activity: what is the person's eye orientation (characterized by the optical axis $\mathbf{o}$) and how are the eyelids open $\{u_e, l_e\}$.

As shown in Figure 6.1, given these parameters, an eye and eyelid configuration can be specified, from which a semantic segmentation of the eye region can be generated. The generative process then further combines this segmentation with session dependent color model distributions, parametrized by $\{\Lambda_l\}_{l=1..3}$, to produce colored eye images.

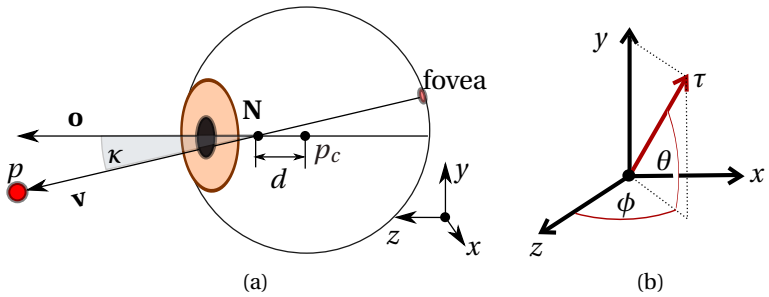


Figure 6.2: (a) Eye geometry with optical (**o**) and visual (**v**) axis definition. (b) spherical parametrization of an arbitrary axis "τ".

The proposed probabilistic model is thus able to compute the likelihood of such eye images, which constitute our observations. Hence, during a training phase, user parameters can be learned by maximizing the likelihood of gaze annotated training samples, while at test time, the image optimization leads to the actual estimation of **m**, and thus the line of sight (*LoS*).

In the following, we describe more precisely the different elements of the G^{3}E model: the eye geometric model, the parametric segmentation function, the definition of the image likelihood, and the generative process.

6.3.2 Eye geometric model

Fig. 6.2a illustrates the geometric eye model we use [Hansen and Ji, 2010]. The process of gazing at a visual target $p \in \mathbb{R}^3$ consists of rotating the eyeball around the point $p_c \in \mathbb{R}^3$ such that the *visual axis* (**v**) intersects p. Notice that both of these quantities are here expressed in the **HCS**, under which p_c is a constant.

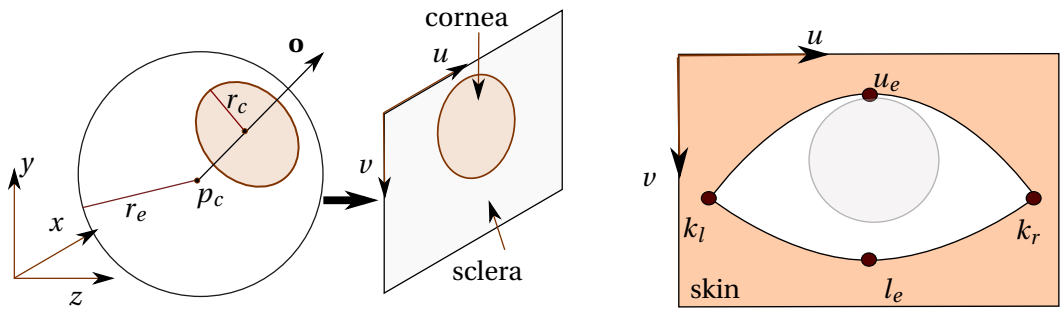
As seen in Figure 6.2a, the *visual axis* is the line connecting the fovea (the point of highest visual acuity in the retina) and the nodal point **N**. It differs from the *optical axis* (**o**), which is the line connecting the center of rotation p_c and the pupil center.

We parametrize each of the **v** and **o** axes by two angles representing the axis elevation and yaw angles[1] (as in Fig. 6.2b). As the eye is a rigid body, the angular difference between these axis is fixed and can be represented by the person dependent angles $\kappa = [\phi_\kappa, \theta_\kappa]^\top$:

$$\mathbf{v} = \mathbf{o} + \kappa \tag{6.1}$$

Therefore, if the person specific *axial parameters* $\mathbf{a} := \{\kappa, d\}$ are known, the eye rotation,

[1] This representation ignores eye torsion. Even though it is known that the eyes rotate according to Listing's and Donder's laws, this [Tweed and Vilis, 1990] simplification was shown to have little impact on gaze estimation as argued by Guestrin and Eizenman [2010].



(a) (b)

Figure 6.3: Eye image parametric segmentation. (a) Cornea-sclera segmentation. (b) Skin region segmentation.

equivalent to **o**, can be defined as a function of the position of p. We denote this process as:

$$\mathbf{o}(p) = [f_\phi(p;\kappa,d,p_c), f_\theta(p;\kappa,d,p_c)]^\top \tag{6.2}$$

The derivation of f_ϕ and f_θ can be found in Appendix B.1.

6.3.3 Parametric segmentation function

In our model an eye image is segmented into three regions: the cornea[2], sclera and skin. Figure 6.3 shows in detail our parametric segmentation: assuming that the user eye geometric parameters $\mathscr{U}$ are known, then a given eye orientation **o** defines a cornea-sclera segmentation, obtained as the orthographic projection of the 3D cornea contour (limbus) into the xy plane of the **HCS**, followed by a transformation to image coordinates uv. This is possible due to the eye image pose-rectification procedure described in Sec. 6.2 which well defines the mapping from the 3D geometry, with respect to the **HCS**, into the pose-rectified eye image coordinates.

To define the segmentation of the skin region, we rely on a set of parameters characterizing the eyelids structure (eye corners k_l and k_r) and another set controlling the eyelids opening. We take a simple approach, shown in Figure 6.3b, where the upper and lower eyelids are quadratic bezier curves sharing the eyelids corners k_l and k_r.

The vertical position of the inner control points are denoted as u_e and l_e, while the horizontal position of the inner control points is given by the horizontal average of the eye corners. These points define the eyelids opening, and thus, the skin segmentation. The skin class overrides the sclera and cornea regions in the overall segmentation.

Given this procedure, we define the *segmentation function* given in Eq. 7.5. Note that this is

[2]We define here "*cornea*" as the 2D region surrounded by the limbus and composed of the pupil and iris regions.

also a function of the parameters which define the geometric structure of the eyes and the current movement of the eye and eyelids.

$$S_{u,v}(l;\mathbf{m},\mathbf{s},p_c) = \begin{cases} 0 & \text{if pixel } (u,v) \notin \text{class } l \\ 1 & \text{if pixel } (u,v) \in \text{class } l \end{cases} \tag{6.3}$$

The derivation of the segmentation process is described in detail in Appendix B.2.

6.3.4 Image likelihood and outlier modeling

So far we used 3 classes to define the eye image segmentation regions. Here we introduce a fourth class to model pixel outliers, denoted by the variable $\lambda = \{0,1\}$, where 1 indicates that the pixel is an outlier. This is intended to address missing data, occlusions and specular reflections.

Our observation data is an eye image $\mathbf{I}$ (pose-rectified). Its likelihood, given the parameters, is defined as $p(\mathbf{I}|.) = \prod_{u,v} p_{u,v}(c|.)$ which assumes that pixels (the c values) are independent observations given the parameters. To model the likelihood of individual pixels, we define the color distribution associated to a class l as $p(c|\Lambda_l)$, a 2 component GMM in the RGB space.

For outliers we assume an equal probability of observing any color, such that $p(c|\lambda = 1) = \epsilon$. The likelihood of a color pixel is then simply defined as the likelihood given its class (either an outlier, or one of the 3 eye region classes), which can be written in condensed form as:

$$p_{u,v}(c|\lambda,\mathbf{m},\mathbf{s},p_c,\{\Lambda_l\}_l) = \epsilon^{\lambda}\left[\prod_l p(c|\Lambda_l)^{S_{u,v}(l;\mathbf{m},\mathbf{s},p_c)}\right]^{1-\lambda} \tag{6.4}$$

6.3.5 Generative model

The graphical model of our geometric generative gaze estimation approach is shown in Figure 6.4. It is a stochastic extension of the full process of gazing up to the generation of eye images, under which every geometric parameter is defined as a random variable.

Let us denote by $x \sim \mathcal{N}(\mu_x,\sigma_x)$ a random variable x being drawn from a univariate Gaussian distribution with mean μ_x and standard deviation σ_x, and the "hat" ($\hat{.}$) notation to represent the hyperparameters of a prior distribution, e.g., $\hat{x} := (\hat{\mu}_x,\hat{\sigma}_x)$. The generative process shown in Figure 6.4 can be described as follows:

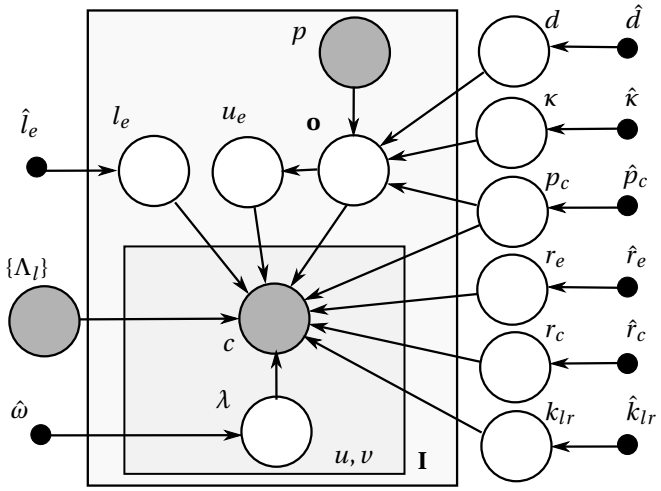


Figure 6.4: Representation of the geometric generative gaze model as a probabilistic graphical model. The symbols are described in Table 6.1.

- Draw the eyeball rotation center p_c:
  - $p_c \sim \left(\mathcal{N}(\hat{\mu}_{p_{cx}}, \hat{\sigma}_{p_{cx}}), \mathcal{N}(\hat{\mu}_{p_{cy}}, \hat{\sigma}_{p_{cy}}), \mathcal{N}(\hat{\mu}_{p_{cz}}, \hat{\sigma}_{p_{cz}})\right)$
- Draw axial parameters $\mathbf{a} := \{\kappa, d\}$:
  - $\kappa \sim \left(\mathcal{N}(\hat{\mu}_{\phi_\kappa}, \hat{\sigma}_{\phi_\kappa}), \mathcal{N}(\hat{\mu}_{\theta_\kappa}, \hat{\sigma}_{\theta_\kappa})\right)$
  - $d \sim \mathcal{N}(\hat{\mu}_d, \hat{\sigma}_d)$
- Draw "structure" parameters $\mathbf{s} := \{r_e, r_c, k_l, k_r\}$:
  - $r_e \sim \mathcal{N}(\hat{\mu}_{r_e}, \hat{\sigma}_{r_e})$
  - $r_c \sim \mathcal{N}(\hat{\mu}_{r_c}, \hat{\sigma}_{r_c})$
  - $k_l \sim (\mathcal{N}(\hat{\mu}_{k_{lu}}, \hat{\sigma}_{k_{lu}}), \mathcal{N}(\hat{\mu}_{k_{lv}}, \hat{\sigma}_{k_{lv}}))$
  - $k_r \sim (\mathcal{N}(\hat{\mu}_{k_{ru}}, \hat{\sigma}_{k_{ru}}), \mathcal{N}(\hat{\mu}_{k_{rv}}, \hat{\sigma}_{k_{rv}}))$
- For each image $I = 1, \ldots, N$:
  - Draw the visual target $p \sim uniform$
  - Draw movement parameters $\mathbf{m} := \{\mathbf{o}, u_e, l_e\}$:
    - $\mathbf{o} \sim \left[\mathcal{N}(f_\phi(p; \mathbf{a}, p_c), \hat{\sigma}_\mathbf{o}), \mathcal{N}(f_\theta(p; \mathbf{a}, p_c), \hat{\sigma}_\mathbf{o})\right]^\top$
    - $u_e \sim \mathcal{N}(a_u\theta + b_u, \hat{\sigma}_{u_e})$
    - $l_e \sim \mathcal{N}(\hat{\mu}_{l_e}, \hat{\sigma}_{l_e})$
  - For each $(u, v) = [1, \ldots, width], [1, \ldots, height]$:
    - Draw outlier or not indicator $\lambda \sim Bernoulli(\hat{\omega})$
    - Draw pixel color $c \sim p_{u,v}(c|\lambda, \mathbf{m}, \mathbf{s}, p_c, \{\Lambda_l\}_l)$

It is important to make a few remarks about this model:

- **Upper eyelid opening**. The upper eyelid is correlated with the elevation angle of the eye by means of a linear Gaussian model. This encodes the effect of the upper eyelid following the vertical orientation of the eye.

- **Eye rotation (optical axis)**. A stochastic extension of Eq. 6.2 was defined to allow uncertainty in the target position or eye fixation.

- **Stochastic segmentation**. Under this model the segmentation becomes a stochastic process. Therefore, drawing a sample from the geometric parameters, or the movement parameters **m**, is equivalent to "drawing a segmentation".

- **Prior distributions and hyperparameters**. Prior distributions have a semantic and/or anatomical interpretation. Therefore the hyperparameters are fixed to values that can be found in the literature (e.g., $r_e \approx 12mm$) or are a consequence of the pose-rectification processing described in Section 6.2 (e.g., it is known where the eye corners are expected to be from the eye image cropping from the 3DMM fitting).

- **Color distributions**. In this thesis, the color model parameters $\{\Lambda_l\}$ are defined as observed. In practice, we acquire color samples from a single image to estimate them. Automatic color model learning is left for future work. Notice that decoupled color modeling is an important advantage of G^{3}E. It allows for adaptation to different illumination and contrast conditions, without the need to re-estimate the geometric parameters.

6.4 Model inference

There are two inference goals for the G^{3}E model:

- **Training phase**. Provided a set of pairs of image samples and visual target locations we aim to infer the person dependent geometry $\mathscr{U}$.

- **Test phase**. Given an input image, we infer **m**, *i.e.*,the eye rotation **o** and eyelids opening leveraging on the previous estimation of $\mathscr{U}$. The inferred **o** is used to estimate the direction of the visual axis (cf. Equation 6.1). The 3D line of sight is defined by referring the in-eye rotated **N** and the visual axis **v** to the **WCS**.

In this thesis we resort to Variational Bayes (VB) as an approximate inference method to address the complexity of our model. We summarize the main points below. A more detailed description can be found in Appendix B.3.

6.4.1 Variational Bayes

Let $\mathbf{X}$ denote the observed data and $\mathbf{Z}$ to be the latent variables to infer (e.g., the geometric parameters). In VB the posterior $p(\mathbf{Z}|\mathbf{X})$, which might not be possible to obtain analytically, is approximated by some *proposal distribution* $q(\mathbf{Z})$. This leads to the definition of the *variational lower bound* $\mathscr{L}(q)$, a functional whose maximization with respect to the function q is equivalent to a minimization of the Kullback-Leibler divergence between $q(\mathbf{Z})$ and $p(\mathbf{Z}|\mathbf{X})$. Therefore, the optimal $q^*(\mathbf{Z})$ is then used as a substitute of the unknown posterior $p(\mathbf{Z}|\mathbf{X})$.

6.4.2 Proposal distribution

As described in Appendix B.3 an iterative procedure may be derived to obtain $q^*(\mathbf{Z})$, where the iterative updates are obtained, in some cases, analytically, in close-form. This depends on assumptions made about $q(\mathbf{Z})$ (e.g., $q(\mathbf{Z})$ is factorized), and the functional form of the conditional distributions (e.g., using conjugate priors). However, this is not the case for our proposed model, due to the complex relations induced by f_ϕ, f_θ and S (cf. Eq. 6.2 and Eq. 7.5).

Instead, we propose to define $q(\mathbf{Z})$ with the following *parametric* form:

$$
\begin{aligned}
q(\mathbf{Z}) = {} & \mathcal{N}(\mu_d, \sigma_d)\mathcal{N}(\mu_{\phi_\kappa}, \sigma_{\phi_\kappa})\mathcal{N}(\mu_{\theta_\kappa}, \sigma_{\theta_\kappa})\mathcal{N}(\mu_{r_e}, \sigma_{r_e})\mathcal{N}(\mu_{r_c}, \sigma_{r_c})\mathcal{N}(\mu_{p_{cx}}, \sigma_{p_{cx}}) \\
& \mathcal{N}(\mu_{p_{cy}}, \sigma_{p_{cy}})\mathcal{N}(\mu_{p_{cz}}, \sigma_{p_{cz}})\mathcal{N}(\mu_{k_{lu}}, \sigma_{k_{lu}})\mathcal{N}(\mu_{k_{lv}}, \sigma_{k_{lv}})\mathcal{N}(\mu_{k_{ru}}, \sigma_{k_{ru}})\mathcal{N}(\mu_{k_{rv}}, \sigma_{k_{rv}}) \\
& \prod_I \left[\mathcal{N}(\mu_\phi, \sigma_\phi)\mathcal{N}(\mu_\theta, \sigma_\theta)\mathcal{N}(\mu_{u_e}, \sigma_{u_e})\mathcal{N}(\mu_{l_e}, \sigma_{l_e}) \prod_{u,v} q(\lambda)\right],
\end{aligned}
\tag{6.5}
$$

where we omit the image and pixel indices to avoid clutter.

In Equation 6.5 every continuous random variable has been defined as a univariate Gaussian. The motivation to propose this parametric $q(\mathbf{Z})$ is that finding $q^*(\mathbf{Z})$ is equivalent to finding the optimal set of Gaussian parameters (means and standard deviations). Therefore, aiming to optimize $\mathscr{L}(q)$, with respect to $q(\mathbf{Z})$, we can compute the derivatives of $\mathscr{L}(q)$ with respect to the Gaussian distribution parameters. Furthermore, inspired by the work of Opper and Archambeau [2009], we can address the complex relations of the model by computing these derivatives from Monte Carlo expectations[3].

A factorized $q(\mathbf{Z})$ also allows to optimize $\mathscr{L}(q)$ in an iterative fashion, where one factor is optimized at the time, leading to an increase of $\mathscr{L}(q)$ until global convergence.

The only non continuous variable is λ. It can be shown that the optimal $q(\lambda)$ in this model is a Bernoulli distribution with $P(\lambda = 1) = \omega$, where ω is given by

$$
\omega = \frac{\hat{\omega}}{(1-\hat{\omega})\frac{1}{\epsilon}\prod_l p(c|\Lambda_l)^{\mathbb{E}_{\mathbf{m},\mathbf{s},p_c}[S_{u,v}(l;\mathbf{m},\mathbf{s},p_c)]} + \hat{\omega}}
\tag{6.6}
$$

[3] All expectations are defined with respect to the current estimate of $q(\mathbf{Z})$

Algorithm 4 Geometric generative gaze model inference.

Set initial $q(\mathbf{Z})$ from the prior distribution parameters.
repeat
- Optimize $\mathscr{L}$ w.r.t. eye corners and all eyelids opening:
  $q(k_{lu})q(k_{lv})q(k_{ru})q(k_{rv})\prod_I q(u_e^I)q(l_e^I)$
- Optimize $\mathscr{L}$ w.r.t. eyeball geometry and orientation:
  $q(r_e)q(r_i)q(p_{cx})q(p_{cy})\prod_I q(\mathbf{o}^I)$
- Optimize $\mathscr{L}$ w.r.t. axial parameters and eyeball depth:
  $q(\mathbf{a})q(p_{cz})$
- Update outliers $q(\lambda^I_{u,v})$ for all pixels using Eq. 6.6

until Convergence
Return $q^*(\mathbf{Z})$

The derivation of Eq. 6.6 can be found in Appendix B.4. Notice that $\mathbb{E}_{\mathbf{m},\mathbf{s},p_c}\left[S_{u,v}(l;\mathbf{m},\mathbf{s},p_c)\right]$ can be interpreted as the *expected segmentation* of an image. Therefore, we can interpret the result of Equation 6.6 as follows: the color of a pixel is identified as an outlier if, either it is unlikely for any class according to the parameters $\{\Lambda_l\}$, or that it can be likely for a given class according to the parameters $\{\Lambda_l\}$, but is spatially incoherent for that class, according to the current belief on the geometric configuration.

6.4.3 Efficient group factor optimization

We can optimize $\mathscr{L}$ efficiently by defining Jacobians over groups of variables, for example $\mathbf{J_a} = [\frac{\partial\mathscr{L}}{\partial\mu_{\phi_\kappa}}, \frac{\partial\mathscr{L}}{\partial\sigma_{\phi_\kappa}}, \frac{\partial\mathscr{L}}{\partial\mu_{\theta_\kappa}}, \frac{\partial\mathscr{L}}{\partial\sigma_{\theta_\kappa}}, \frac{\partial\mathscr{L}}{\partial\mu_d}, \frac{\partial\mathscr{L}}{\partial\sigma_d}]^\top$. This is efficient in terms of derivatives computation, as we found that their Monte Carlo expectations require group sampling rather than univariate sampling, due to the inter-parameters dependencies induced by Eq. 6.2 and Eq. 7.5.

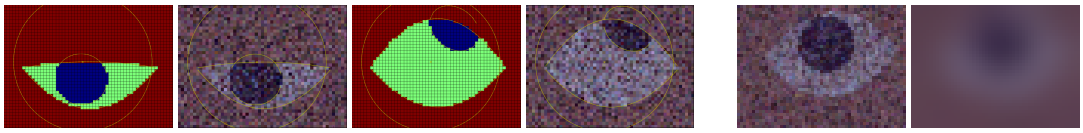
Gradient ascent is then used to find the optimal Gaussian parameters of the corresponding factor of $q(\mathbf{Z})$ (e.g., $q(\mathbf{a})$).

6.4.4 Inference algorithm

Training. Our overall inference method is given in Algorithm 4. This method finds the person-specific geometry from a set of eye images and their corresponding p.

Test phase (Gaze inference). At test time, the geometry is fixed and we only optimize with respect to the test image's $q(\mathbf{m})$ and outliers factor in an iterative fashion. The main difference with the training phase is that, in this case, the visual target location p is unknown. Furthermore, as we assume a uniform prior over p, its influence on the estimation of the optical axis becomes uninformative.Therefore, the terms related to p can be ignored in the optimization.

The mode of $q^*(\mathbf{Z})$ can be used as the MAP estimate of $\mathbf{m}$, from which we can compute the visual axis, leading to the definition of the 3D *LoS* for the given test image.



(a) (b)

Figure 6.5: (a) Synthetic data samples (drawn segmentation and the generated image from color sampling). (b) Sample (left) smoothed by a gaussian filter of $\sigma = 3.0mm$ (right).

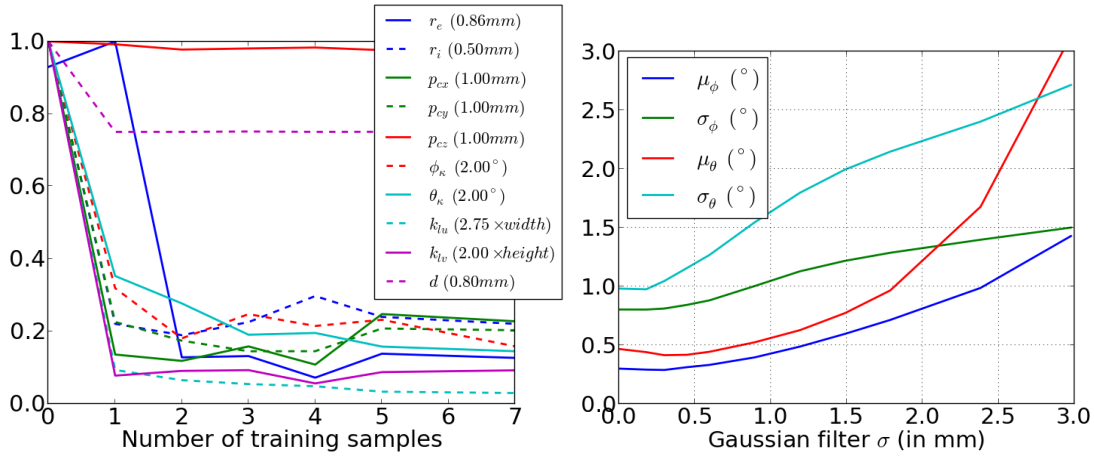


Figure 6.6: Left. Parameter estimation error vs. number of eye training samples. The y axis scale is given in the legend of each parameter. Right. Mean and standard deviation (derived from the inferred $q(\mathbf{o})$) of the gaze estimates $\mathbf{o} := (\phi, \theta)$, vs. the standard deviation of the Gaussian blurring filter ($1mm = 1.68pixels$). For the gaze means, the deviation from their true values is plotted. For each experiments, averages over 500 runs are reported.

6.5 Experiments

To validate our model, we first studied its behavior using synthetic data. We then compared it against a representative geometric based method and an appearance based approach on real data from the EYEDIAP database to validate its advantages and properties.

6.5.1 Experiments on synthetic data

To validate our method, we created synthetic data using the generative process described in Section 6.3.5. Examples are shown in Figure 6.5a; their resolution in pixel is 55×40. Synthetic data allows us to study the inference scheme and the observability of the gaze model parameters by comparing the parameters inferred by G^{3}E to their true values.

The left plot of Figure 6.6 shows the parameter estimation errors as a function of the number of training samples, where each parameter is inferred separately while the other parameters are set to their true values. We can conclude the following: i) almost all parameters can be well

estimated, and this requires only a few samples; ii) d and the eyeball depth p_{cz} are difficult to infer, due to their small impact on $\mathbf{o}$. Nevertheless, this means that their impact on gaze estimation is small. iii) The visual axis deviation angle parameters (κ), which are important for accurate gaze estimation but are often neglected, are well constrained by the image likelihood and the known object position p, and can thus be inferred.

The right plot of Fig. 6.6 shows a similar experiment: we evaluate the gaze estimation accuracy (assuming the true person specific geometric parameters are known) as a function of image resolution simulated through blurring (see Fig. 6.5). Notice the high robustness with respect to resolution due to the optimization of a global image likelihood measure. We also show the estimated variances, which correctly reflect the uncertainty of the gaze estimates.

These results suggest that G^{3}E may potentially provide not only the estimated gaze, but also the uncertainty on the estimate. Furthermore, given proper training data, our approach may achieve highly accurate gaze estimation ($< 2°$ error) at test time under poor sensing conditions.

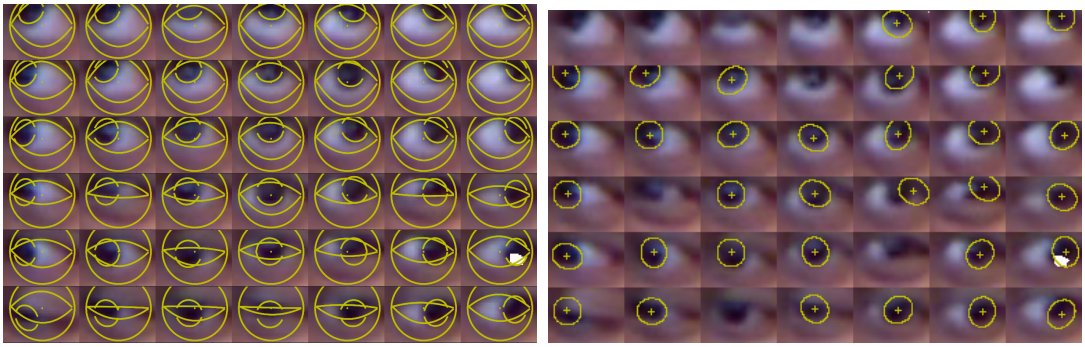
6.5.2 Real data evaluation

To evaluate the G^{3}E model we used recording sessions from the EYEDIAP database, which is well described in Chapter 4. However, at the time these experiments were conducted, the EYEDIAP database was at an early stage of development, far from the version made publicly available. Therefore, the experiments described in this Section do not follow the proposed benchmarks, and some of the frames validity checks (described in Section 4.4.2) were not conducted due to, e.g., lack of manual annotations. Therefore, the obtained errors are higher than in comparison to the experimental results obtained in Chapter 5. Direct comparisons to the methods of Chapter 5 using the EYEDIAP benchmarks are left for future work[4]. Nevertheless, the experiments presented in the following sections are intended to validate specific characteristics of the G^{3}E model which are advantageous with respect to the appearance and geometric based paradigms.

We used recording sessions involving either the *FT* (floating target) or the *CS* (continuous screen) visual targets under a static (*SP*) or moving head pose (*MP*). Recall that the distance of the participant to the screen and sensor was $\approx 85cm$. In the experiments using the *FT* visual target, the participant's distance to the sensor was approximately 1.2m, resulting in eye image sizes between 20×14 (*CS*) and 13×9 (*FT*). In these experiment, the pose-corrected image are upsampled to a fix resolution of 55×40 pixels with known pixel size ($0.595mm/pixel$).

Notice that, unless stated otherwise, the G^{3}E model is trained from 42 gaze annotated eye image samples. These samples are collected as a structured training set (cf. Section 4.4.4) such that the ground truth gaze angles are regularly distributed among the range of observed gaze directions, as illustrated in Figure 6.7a. To learn the color distributions, we manually segment 1 to 3 images, from which we pick color samples of the three classes: cornea, sclera and skin.

[4] These experimental evaluations are indeed amongst our short term goals



(a) (b)

Figure 6.7: (a) Geometric fitting given by G3E. (b) Ellipse fitting given by the Starburst algorithm on training data collected using the floating target.

The inference process is conducted as described in Algorithm 4. The hyperparameters, i.e., the prior distributions, were set manually. Anatomical values were found in the literature, e.g., [Hansen and Ji, 2010], in millimeters: $\hat{\mu}_{r_e} = 12$, $\hat{\mu}_{r_c} = 5.5$, $\hat{\mu}_d = 7$ and $\hat{\mu}_{p_c} = \mathbf{o^h}$ (eyeball position in the **HCS**[5]), and their standard deviations were set to 0.8. The angular values were $\hat{\mu}_{\theta_\kappa} = -1.5°$ and $\hat{\mu}_{\phi_\kappa} = \pm 5°$ (the sign is different for the left and right eye [Guestrin and Eizenman, 2006]), and their standard deviation were set to 1°. Finally, the eye corners are defined in the pose-rectified eye images domain. The mean of their priors ($\hat{\mu}_{k_{lr}}$) are set to the projection of the eye corners vertices (obtained from the 3DMM topology) and the standard deviation is set to 0.05 times the image width.

As performance measure, we used the angular gaze error, defined as the angle between the estimated line of sight (LoS) and the vector pointing from the LoS's origin to the (known) visual target's 3D position (p), as described in Section 4.4.7.

6.5.3 G^{3}E inference and geometric methods

We illustrate in Figure 6.7a the inference process output on a set of training samples collected from a recording session involving the *FT* visual target. The result of the training can be visualized using the mode of $q^*(\mathbf{Z})$ (MAP estimate) and by overlaying the contours of the associated segmentation and eyeball structure, as shown in Fig. 6.7a. Our method follows properly the position of the eyelids and eye orientation despite the low resolution and the sometimes unclear boundaries between eye regions.

As a qualitative comparison, we tested the Starburst algorithm [Li et al., 2005] on the same data. This approach is a well known method and representative of the feature extraction step (pupil/iris center localization) in the geometric based paradigm. The Starburst algorithm is based on the fitting of an ellipse to the pupil/iris contour based on thresholded gradient

[5] In practice, we use a coordinate system positioned in $\mathbf{o^h}$ (without rotation), such that $\hat{\mu}_{p_c} = \mathbf{0}$

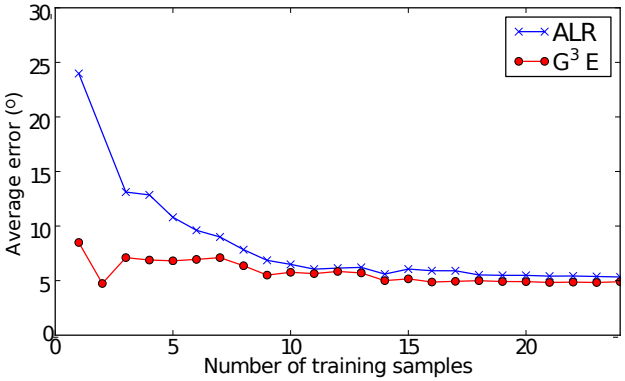


Figure 6.8: Average gaze error as a function of the number of training samples. Computed on test data from a participant gazing at a floating target with a fixed frontal pose.

features, found along rays from an initial estimation of the iris region center. Further features are extracted by searching in the same manner from the features found in the previous step. An iterative process, and the usage of the random sampling and consensus (RANSAC) algorithm makes the ellipse fitting less sensitive to wrong initializations and feature outliers. This method can be seen as an active shape model [Cootes et al., 1995].

In these experiments, we started the starburst fitting process from the true iris center value, as a weakly supervised strategy. Nevertheless, despite this, and tuning the algorithm parameters, we normally obtained results as shown in Figure 6.7b. The low recall and unaccurate estimation demonstrate the important difficulties of ellipse fitting, which is a critical step for many geometric gaze estimation methods, e.g., [Ishikawa et al., 2004, Xiong et al., 2014]. Notice that our approach does not have this limitation as it avoids local feature computations.

6.5.4 G^{3}E and appearance based methods

We compared the G^{3}E model to the method we proposed in [Funes Mora and Odobez, 2012]. As we described in Chapter 5, this method also builds over the head pose invariant gaze estimation approach. From the pose-rectified eye images, the gaze direction is estimated using adaptive linear regression (ALR) from a set of gaze annotated training samples. Notice that in Chapter 5 we empirically found that ALR, originally proposed by Lu et al. [2011a], performs best among different appearance based gaze estimation methods in a supervised setting, when using a small set of training samples. For ALR we typically used the same 42 training samples, organized as 7 gaze yaw angles and 6 gaze elevation angles. Our intention is to contrast to the appearance based paradigm. We now describe the result of experiments designed to raise awareness of the limitations of appearance based methods.

Number of training samples

As concluded in Section 6.5.1 our model is adequate for training from few data. We validated this on data from the EYEDIAP database (*FT* visual target), and compared to ALR, as shown

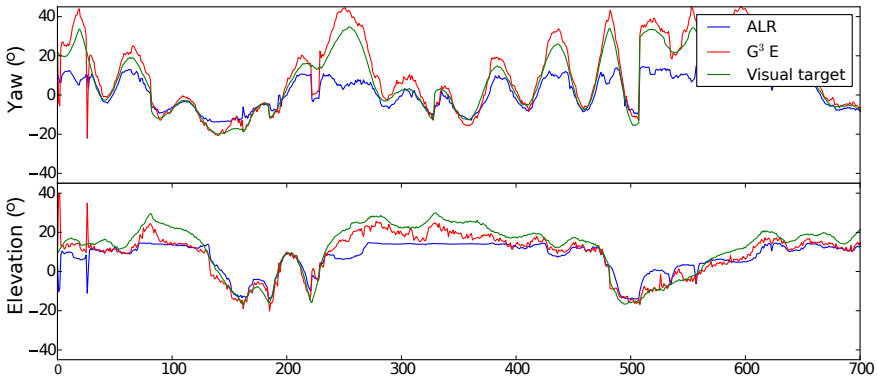


Figure 6.9: Estimated eye rotation (°) on a test sequence, with training samples restricted to the $[-15°, 15°]$ range.

in Figure 6.8, which presents a typical error curve obtained for a participant in functions of the number of training samples. As shown in Figure 6.8, ALR, as any other appearance based methods, need to cover densely the gaze space with the training samples in order to achieve lower errors. However, G^{3}E, even from a few observations is already capable to estimate gaze with an acceptable performance.

Gaze extrapolation

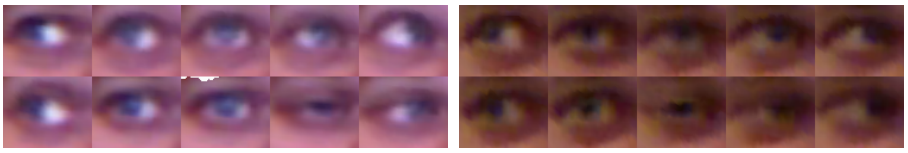
As our method is based on a explicit eye model, we argue it can extrapolate to gaze directions outside the training set. To illustrate this, we conducted an experiment where we collected the training samples restricted to gaze yaw and elevation angles within the range $[-15°, 15°]$ to train the ALR and G^{3}E models. Notice that this range of gaze directions is typical of a scenario involving a gaze calibration procedure using a computer screen, to display visual targets.

Figure 6.9 shows the gaze tracking results on a test sequence. We can observe that this claim is validated. ALR, as any interpolation based method, is not able to estimate gaze outside the range of gaze directions used for training, thus causing the saturations observed in Figure 6.9. On the contrary, our method correctly extrapolates to gaze directions further than the range observed in the training set.

Gaze estimation across different sessions

In this experiment we exploited recorded sessions involving the *FT* visual target and maintaining a static head pose (*SP*), and which were collected for the same participants 6 months apart and in different ambient conditions. Across sessions, denoted A and B, there is a drastic change in the illumination and distance to the camera, as illustrated in Fig. 6.10.

We then trained an ALR and a G^{3}E model using the data from session A. For the G^{3}E approach, applying directly the learned model -including the color distributions from session A- on



(a) (b)

Figure 6.10: Eye image samples across different sessions for the same participant; involving the floating target (*FT*) and static head pose (*SP*) conditions. Participant 1 in (a) session A and; (b) session B.

Table 6.2: Mean angular error (°) when training the model on session A and estimating gaze on session B. See text.

Method	Participant 1	Participant 2
ALR	21.7	25.0
G^3E	8.0	5.5

session B results in large errors (40.2° and 38.2° for the respective participants). This is due to the obvious color mismatch between the two conditions. However, we can easily leverage on the important property of the G^3E model which is the decoupling of the ambient conditions (modelled by $\{\Lambda_l\}$) from the person-specific geometry $\mathscr{U}$. By learning the color distributions of session B, using the same process described in Section 6.5.2, we quickly obtain an adapted model that results in good performance, as shown in Table 6.2.

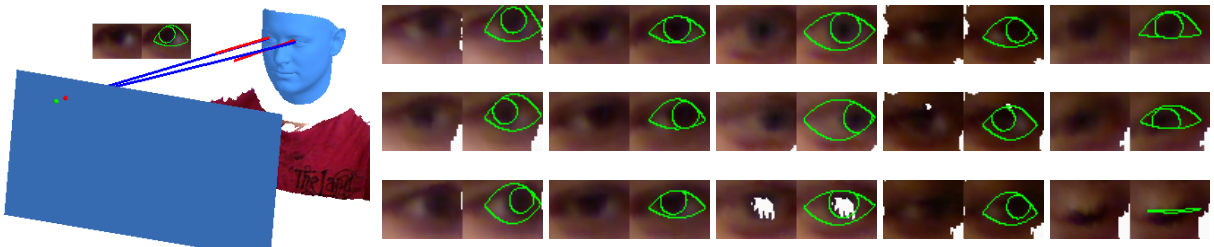
On the other hand, given the lack of geometrical model, ALR does not offer much flexibility for adaptation. Even if it relies on normalized features that should be robust to global illumination variation within the eye image (see Section 5.3.3), the results in Table 6.2 shows that the Session A model is not appropriate for Session B, demonstrating that session changes go beyond simple illumination and contrast corrections.

The automatic learning of color distributions for G^3E is left for our future work, but this experiment already validates its potential for cross-session adaptation.

6.5.5 Screen gazing evaluation

We evaluated the performance of our method for the screen target estimation task, where we considered both the static (*SP*) and moving (*MP*) head pose case for five participants. Due to the proximity to the depth sensor and, as we used the *DDM* pose-rectification procedure (cf. Section 5.2.3), there are regularly missing depth data which leads to missing patches in the pose-rectified eye image, as can be seen in Figure 6.11b. In the G^3E model, we can well address this problem by forcing the pixels to be outliers (i.e., setting $\omega \sim 1$ for missing pixels). As ALR does not provide a straightforward way to handle missing data, we do not report results here.

The results are summarized in Table 6.3. Given the quality of the input data and that head



(a) (b)

Figure 6.11: Screen gaze estimation task. (a) 3D rendered RGB-D frame. The user's face region is replaced by the 3D facial template, rendered with the estimated head pose. The blue lines and green dot on the screen are the ground truth. The red lines correspond to the estimated lines of sight and the red dot is the screen intersection (for the left eye). (b) Test images for the left eye (originally, ≈ 20 pixels eye width, prior to the pose rectification). The data shown at each column correspond to a different participant. White pixels correspond to missing depth data. The contours represent the mode of $q^*(\mathbf{Z})$ (the MAP estimate on the geometry).

Table 6.3: Gaze angular median error (°) for people looking at screen targets.

	Participant					
Head pose	1	2	3	4	5	Avg
Static (*SP*)	2.9	2.7	3.1	2.5	5.9	3.4
Moving (*MP*)	9.3	5.5	3.6	4.6	8.6	6.3

pose variation was within a range of $\pm 30°$ for yaw and elevation, the performance are highly promising. To illustrate this, we provide in Fig. 6.11 an example of the setup together with qualitative segmentation results for the 5 participants.

These results show that our method has a good behavior at test time, although we observe on the bottom right a problematic situation for our approach which is extreme gazing down, where the cornea region gets heavily occluded by the eyelid.

6.6 Conclusions and future work

In this Chapter we have proposed a novel paradigm for gaze estimation. We call this methodology geometric generative gaze estimation (G^{3}E). It is based on a geometric understanding of the 3D gaze action up to the generation of eye images, formalized as a generative process. We developed an inference technique, based on Variational Bayes, to find the person specific geometric parameters from training data (i.e., gaze annotated eye images), and also to estimate the gaze direction and eyelids opening at test time, from the known person specific geometry.

We have demonstrated that this method has many advantages with respect to previous appearance and geometric based methods (resp. ABM and GBM), by conducting experiments on both synthetic and real data. Thanks to the usage of priors on the geometric parameters, G^{3}E

is adequate for training from a few training samples. This is valuable for many applications as it reduces the amount of needed user cooperation.

This model is also capable of gaze extrapolation, i.e., to estimate gaze on samples with further gaze directions than the range observed in training set, which is, otherwise, a known limitation of appearance based methods. This has important practical implications, as it allows to design more flexible calibration sessions, e.g., to calibrate on a screen, but to estimate gaze in the 3D space.

The G^3E model has also been proven to be adequate for low resolution sensing, as it was designed to avoid the detection/tracking of local eye features. This has been, otherwise, a strong requirement for GBM and have therefore restricted their usage to high quality data sensing conditions.

The proposed model correctly decouples the geometric parameters from the appearance parameters. Whereas the geometry is modelled by the person specific geometric parameters, the appearance is modelled as independent color distributions. These color distributions are not only valuable to discriminate between the semantic regions, but also to reflect the sensing conditions which have an impact on the pixel color observations (e.g., illumination). This decoupling enables the adaptation from a previously trained model to different sensing conditions, without the need to re-estimate the person specific geometric parameters.

Finally, in this Chapter, the G^3E model was designed over the RGB-D based framework for head pose invariant gaze estimation. Meaning, the overall approach is head pose invariant, allowing to estimate gaze for unconstrained user movements.

Future work. There are many interesting research directions that may be investigated under this framework. The main element which needs to be addressed is the automatic learning of color distributions. Notice that this task, although not trivial, is less challenging than the inference of the geometric parameters. Different strategies could be exploited: the skin color distributions may be obtained from face regions during tracking; the sclera is known to be always “whiter” in comparison to the “cornea” region, this observation may help to discriminate between these two classes. We could further profit from color distribution priors, trained from collections of eye images. In this manner, the inference of the color distributions parameters could be incorporated within the overall inference scheme. Furthermore, an otherwise cooperative scenario may be designed: assuming the user specific geometric parameters are known, by fixating at a single point we can use the model to generate a segmentation to pick color samples automatically from this single image.

The eye geometric model we used is a simplified model of the human eye. In future work, it might be interesting to consider eye torsion, as its impact may be more significant for larger gaze directions than in the cases where its influence has been neglected [Guestrin and Eizenman, 2010]. In addition, the model’s prior geometric parameters were manually defined. Alternatively, these parameters could be estimated from a collection of high-quality 3D eye

models, which could be retrieved using 3D reconstruction techniques [Bérard et al., 2014].

An interesting research direction, which aims at further reducing the need for user calibration, may profit from the probabilistic formulation of G^{3}E. By leveraging prior research on human attention, we can design inference techniques which are suitable for unsupersived or weakly supervised scenarios. The strong geometric priors of the model may have an important impact in these cases, as they are helpful to regularize the plausible set of geometric solutions.

The derived inference scheme, based on Variational Bayes (VB), is an interesting approach which leads to fast convergence and the further estimation of the parameters uncertainty. However, the model complexity, and the Monte Carlo (MC) based variational bound derivatives computation makes the optimization process not trivial. Notice that, due to the MC sampling, this optimization is actually stochastic. In practice, we sometimes noticed this had a negative impact on the resulting parameters estimates. Therefore, for future research, it would be of value to investigate methods to make the VB optimization more robust, or alternative inference schemes could be applied to the proposed model, e.g., Gibbs sampling, MCMC, or simple grid sampling, etc.

We believe the G^{3}E methodology has a significant potential for gaze estimation in many diverse scenarios in HHI, HRI and HCI applications. To validate this, we evaluated G^{3}E under challenging conditions, such as low resolution imaging, unconstrained user motion (large head pose variations) and minimal user cooperation (reduced number of training samples). Nevertheless, we believe this methodology may also be a powerful approach in more cooperative scenarios, relying on higher quality data. This last point remains to be validated with future experiments, together with a systematic comparison to the results obtained in Chapter 5, using the EYEDIAP database.

7 Gaze Coding in Natural Dyadic and Group Interactions

7.1 Introduction

The understanding of human non verbal behavior and their functions in interactions is important in many fields of study. Indeed, whereas the understanding itself is an active research topic within the fields of psychology and sociology, its exploitation is considered as crucial for computer scientists for the development of newer generations of socially aware human-computer and human-robot interaction systems.

To this end, in recent years there has been a growing interest in the collection and analysis of dyadic and multi-party corpora of human interactions in natural settings. To conduct behavior and social studies using these corpora it is often necessary to *code* the non-verbal cues, which means to obtain a mid-level representation in which it is known who and when performs a given non verbal action. However, due to the difficulty to manually code non-verbal behavior, and gaze in particular, these studies are normally limited in terms of size and quality.

Automatic methods have been proposed for gaze coding (or "visual focus of attention"). However, due to the many difficulties involved in the sensing of gaze, these methods have then been limited to using the head pose as a proxy [Ba and Odobez, 2006, Jayagopi et al., 2012], or to rely on coarse measurements of gaze [Gorga and Otsuka, 2010]. Therefore, in this Chapter we will propose automatic gaze coding solutions based on the methodologies developed in previous chapters.

Our goal is to propose a system capable of gaze coding under natural behavior and non cooperative participants. Natural behavior implies that the motion of the participants is unconstrained, whereas the lack of cooperation means that it is not possible to conduct a gaze calibration session to train person specific gaze models. We will therefore rely on the methods described in Chapter 5, developed to achieve head pose and person invariance to infer the gaze direction[1]. To derive the coding, i.e., associating a gaze direction with looking at a target,

[1] The methodology proposed in Chapter 6 could potentially be used as well in this context. However, the work and experimental validations in this Chapter were made relying on the elements from Chapter 5.

we will further take advantage that the methodology developed in this thesis (gaze estimation and participants tracking) is done in the 3D space. We will thus propose ways to calibrate a setup composed of multiple RGB-D sensors, and then formulate the gaze coding problem as a geometric analysis of the interaction in the 3D space.

The work presented in this Chapter has been mainly developed in the context of the SONVB project[2], which aims at studying non verbal behavior patterns in *real* job interviews situations. The scenario is composed of two participants: an interviewer and interviewee whose behavior is natural and unconstrained. Therefore, in the following, we mainly formulate the problem and validate the proposed system for dyadic interactions. Most elements of this Chapter were published in [Funes Mora et al., 2013], but we present the recent improvements based on the gaze estimation models developed in Chapter 5 (currently under submission [Funes Mora and Odobez, 2015]).

The rest of the Chapter is structured as follows. In Section 7.2 we describe the proposed methodology. Section 7.3 covers the experiments we conducted to validate the proposed system. In Section 7.4 we discuss extensions, in particular, to the multi-party situation. Finally, in Section 7.5 we present a discussion and conclude the Chapter.

7.2 Proposed gaze coding system

In this section we describe the proposed system for dyadic gaze coding. First, we describe the sensors setup and calibration; then, we briefly recall the gaze tracking elements from Chapter 5, which are relevant to this methodology, followed by the gaze coding approach. Finally, we propose a simple procedure to apply the method of Chapter 5 and obtain the person specific eye image alignment parameters.

7.2.1 System setup

For the proposed setup, shown in Figure 7.1, we employed two RGB-D cameras (Microsoft Kinects™) positioned on a table and facing opposite directions, such that there is a camera assigned per participant.

From this diagram, we can define a set of coordinate systems: the world coordinate system (**WCS**) and a camera coordinate system (**CCS**) per RGB-D sensor A or B. Each participant will also have assigned a head coordinate system (**HCS**), which follows the result of the head pose tracking (cf. Section 7.2.2). In the following we will describe a simple procedure to obtain the pose parameters of the two cameras, based on a few assumptions.

We assume the RGB-D sensors are calibrated, meaning that, per-sensor, the intrisic parameters of the RGB and depth cameras are known, together with the relative pose between the RGB-

[2] https://www.idiap.ch/project/sonvb

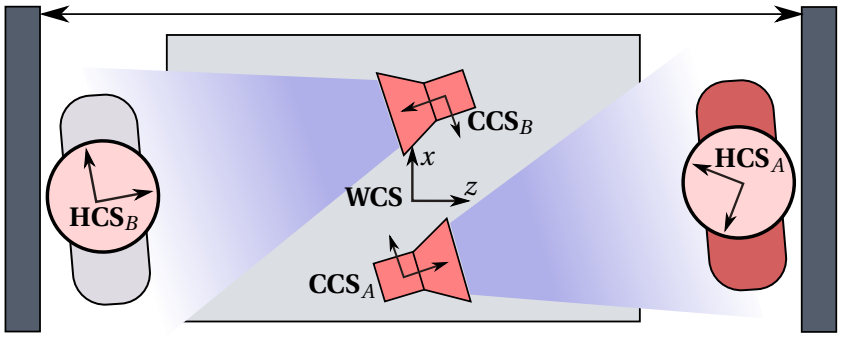


Figure 7.1: Top view of the system setup and definition of the related 3D coordinate systems.

depth stereo pair (extrinsic parameters). These calibration parameters allows us to interpret the RGB-D data as a textured 3D mesh defined with respect to its corresponding camera coordinate system (**CCS**). In our implementation, the **CCS** normally corresponds to the coordinate system of the RGB camera, as defined by the pin-hole camera model.

The 3D pose of each RGB-D camera needs to be estimated with respect to a fixed world coordinate system (**WCS**). This allows to refer all quantities (head pose, gaze, mesh data), initially expressed in their corresponding **CCS**, into the **WCS** and, in this manner, obtain a unified representation, independent of the data source. For example, we can construct a single textured 3D mesh of the overall scene and the interaction, all expressed in the **WCS**.

However, notice that the fields of view of the cameras do not overlap. Thus, we can not use stereo calibration techniques based on 3D correspondences to find their relative pose. We then propose to leverage on the background walls to estimate the camera pose based on the following assumptions, which can easily be met:

1. The wall planes are fronto-parallel;
2. Both cameras are at the same height (e.g., on a table);
3. There is no rotation along the z axis of each camera (roll);
4. The wall-to-wall distance and the distance between cameras are known.

Based on these assumptions, the camera pose estimation procedure works as follows: given a single depth frame we fit a plane to the back wall using the depth pixels segmented based on distance thresholding. Provided that the **WCS** is defined with its z axis perpendicular to the walls (as shown in Fig. 7.1), then the camera's tilt and yaw angles are obtained directly from the wall's normal vector, which corresponds to the **WCS**'s z axis. This assumes the camera roll angle is 0 (assumption 3, above).

The translation of each camera is obtained by first rotating its data to be aligned to the **WCS**. Then, its position along the z axis is obtained from the camera-to-wall distance (from the

fitted plane) and the known wall-to-wall distance. Its position along the y axis is set to 0, as we assume the cameras are at the same height, and finally, the position along the x axis, is set to 0 for one of the cameras, whereas for the other, it is obtained from the measured camera-to-camera distance (considering the difference along the z axis).

Alternatively, some of these parameters can be defined a priori, from the setup design, e.g., by carefully placing the cameras at the same position along x (therefore, the position along x would be 0 for both sensors), or aligning the camera's yaw angle to the wall, such that its value is 0. However, the conditions we propose can easily be met in a standard room, the calibration procedure is simple to implement and it allows for more freedom on the camera's placement.

7.2.2 Head and gaze tracking

To detect gaze coding events it is necessary to track the head pose and the gaze direction of each participant. To this end we build upon the methodology developed in previous chapters of this thesis.

To retrieve the head pose we use the ICP based tracking algorithm described in Section 3.3, which assumes a person specific face model is first obtained by fitting a 3D Morphable Model (3DMM) [Paysan et al., 2009] to RGB-D data, as described in Section 3.2. The estimated head pose is then obtained as $\mathbf{p} = \{\mathbf{R}, \mathbf{t}\}$, i.e., a 3D rotation and translation, which defines the pose of the **HCS**, with respect to the **WCS**.

For the gaze estimation, given the results obtained in Chapter 5, we decided to use the H-SVR gaze estimation model. This was motivated by the results obtained in the experiments evaluating person invariant gaze estimation algorithms. The H-SVR model either performed very comparable to the best approach, in the *FT* settings, or had the best performance by a larger gap, in the *CS* settings. As part of the contributions of Chapter 5, this approach builds over the head pose invariant appearance based gaze estimation framework. Therefore, it is adequate for this task, as it allows for unconstrained motion from the participants.

Recall that we know an approximate location (per eye) of the eyeball center "$\mathbf{o}^{\mathbf{h}}$", defined with respect to the **HCS**. The output of the gaze algorithm is the vector $\mathbf{v}^{\mathbf{h}}$ also expressed in the **HCS**. To transform these values to the **WCS** it is only necessary to apply the rigid transformation given by the head pose, i.e., $\mathbf{o}^{\mathbf{wcs}} = \mathbf{R}\mathbf{o}^{\mathbf{h}} + \mathbf{t}$ and $\mathbf{v}^{\mathbf{wcs}} = \mathbf{R}\mathbf{v}^{\mathbf{h}}$.

For this setup, we will also consider obtaining a person specific eye alignment, using the synchronized delaunay implicit parametric alignment (SDIPA) approach, which we proposed in Section 5.4.2. In Section 7.2.4 we will propose an approach to easily extract the data required to infer this alignment.

7.2.3 Gaze event detection

To infer whether a person is looking at a visual target, two main elements are needed: the 3D gaze direction and the target position, both referred to the same coordinate system. In dyadic interactions, the visual target $\mathbf{y} \in \mathbb{R}^3$ is mainly the other participant. We aim at discriminating between looking at the other person or not. We will here make the assumption that the point of interest, when looking at the other participant, is between the eyes, such that:

$$\mathbf{y} = \frac{\mathbf{o}^{\mathbf{h}}_{left} + \mathbf{o}^{\mathbf{h}}_{right}}{2}, \tag{7.1}$$

and is defined with respect to the **HCS**. The gaze direction and the head pose are then estimated as described in Section 7.2.2. All quantities can be referred to the **WCS** or any **HCS**, using the system geometry described in Section 7.2.1 and shown in Fig. 7.1.

We can proceed to define the gazing event decision function. Thanks to the unified 3D geometry of the fully calibrated system, we can detect these events as the moments in which the participant's 3D gaze vector intersects the visual target position.

Assume we aim at detecting if the participant A is gazing at participant B. Given that the estimated head poses of the participants are $\mathbf{p}_A := \{\mathbf{R}_A, \mathbf{t}_A\}$ and $\mathbf{p}_B := \{\mathbf{R}_B, \mathbf{t}_B\}$ (referred to the **WCS**), the point of interest at participant B, with respect to his/her **HCS**, is $\mathbf{y}_B$ (cf. Equation 7.1), and the line of sight of the participant A, defined with respect to his/her **HCS**, is $LoS_A := \{\mathbf{o}^{\mathbf{h}}_A, \mathbf{v}^{\mathbf{h}}_A\}$. Then, we define the *gaze reference vector*, expressed in the **HCS** of participant A, as the *unitary* vector $\hat{\mathbf{v}}^B_A$, which points from the eyeball of participant A to the participant B as follows:

$$\hat{\mathbf{v}}^B_A \propto \mathbf{y}^{\mathbf{HCS}_A}_B - \mathbf{o}^{\mathbf{h}}_A = \mathbf{R}^\top_A \left(\mathbf{R}_B \mathbf{y}_B + \mathbf{t}_B\right) - \mathbf{R}^\top_A \mathbf{t}_A - \mathbf{o}^{\mathbf{h}}_A, \tag{7.2}$$

where $\mathbf{y}^{\mathbf{HCS}_A}_B$ is the position of the participant B referred to the **HCS** of participant A. Finally, we can define the gazing decision function as:

$$\arccos\left(\mathbf{v}^{\mathbf{h}}_A \cdot \hat{\mathbf{v}}^B_A\right) < \tau, \tag{7.3}$$

where τ is the gazing angular threshold. Although these elements are referred to the **HCS** of participant A, the decision on the coordinate system to refer all variables does not have an impact on the end result. Notice that we only use the LoS of one of the eyes. In practice, the eye which is more visible to the camera is used in our experiments, which is derived from the estimated head yaw angle.

7.2.4 Eye image alignment estimation

As proposed in Chapter 5, Section 5.4.2, the estimation of person specific eye image alignment parameters can further boost the accuracy of person invariant gaze estimation models. Moreover, it can help to correct for a systematic gaze estimation bias, which can otherwise degrade the gaze coding accuracy. Here we propose a simple procedure to estimate this person specific eye image alignment.

Notice that, at all times during the interaction, the gaze reference vector is well defined, as it only depends on the head pose tracking (as shown in Equation 7.2). If, at a given moment, it is known that the participant A is gazing at participant B, we can collect a gaze annotated sample as $\{\mathbf{I}^R, \hat{\mathbf{v}}_A^B\}$, where $\mathbf{I}^R$ is the pose-rectified eye image (see Figure 5.2).

In practice, this can be done with a minor effort. An experimenter, i.e., a person which requires to extract the gaze coding from the recordings of an interaction, can annotate a few frames in which this event occurs. In Chapter 5 we demonstrated empirically than only a few samples (≈ 5) are enough for this purpose.

Notice that this is not a requirement. Good performance can already be obtained from the person invariant gaze model alone. Nevertheless, as it will be show in Section 7.3, the alignment indeed improves the gaze coding accuracy.

7.3 Experiments

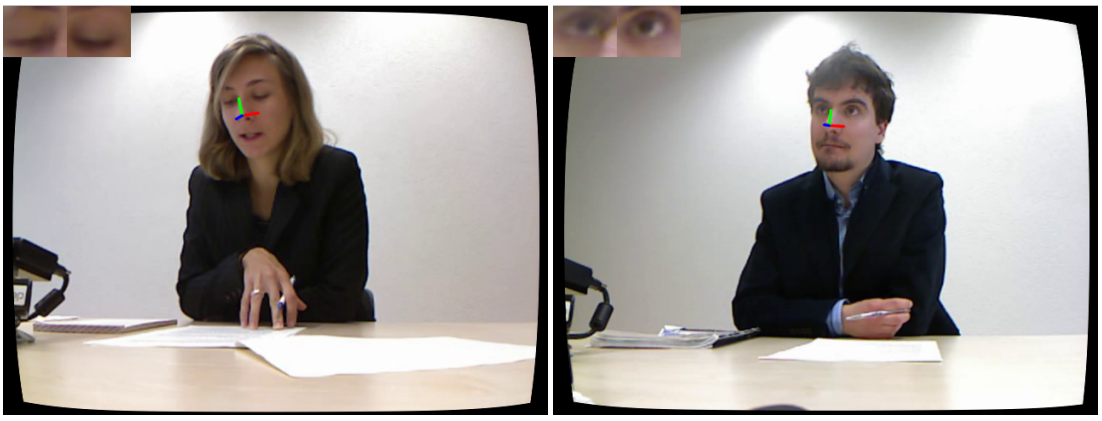
In this section we describe the experiments we conducted to validate the gaze coding system we developed. We first describe the data we used, including its annotations, followed by the experimental protocol and finally we present and discuss the results we obtained.

7.3.1 Data

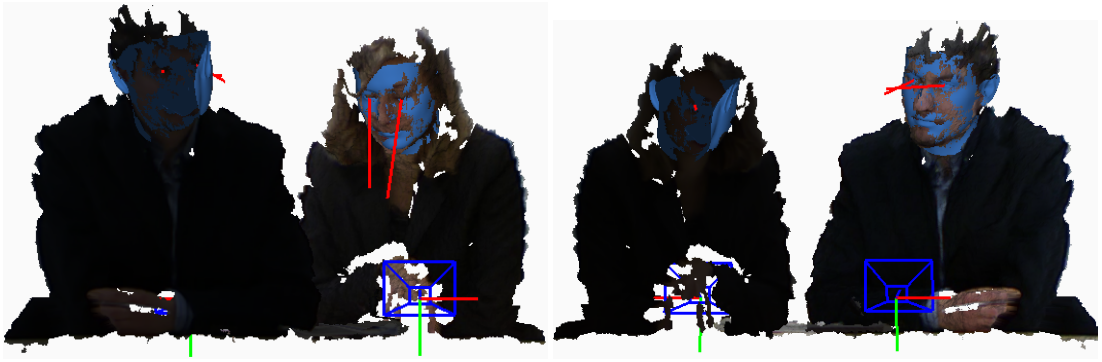
As mentioned in Section 7.2.2, we here use the H-SVR based gaze estimation algorithm. This model was trained from all the participants of the EYEDIAP database (cf., Chapter 4) to obtain a person invariant gaze estimation model, as described in Chapter 5.

For the gaze coding evaluation, we considered five natural dyadic interactions consisting of real job interviews conducted in the context of the SONVB project. One participant has the role of the *interviewer*, while the other one is the *interviewee*. The interviewer was always the same person, but the interviewee was a different subject per interview. Each interview was recorded using the setup as described in Section 7.2, which was calibrated accordingly.

In order to train and evaluate our automatic gaze coding system, one person manually annotated gazing events for both protagonists of the interaction. Gazing events were defined as the time periods in which the person of interest was looking at the other subject. We did not filter



(a)



(b)

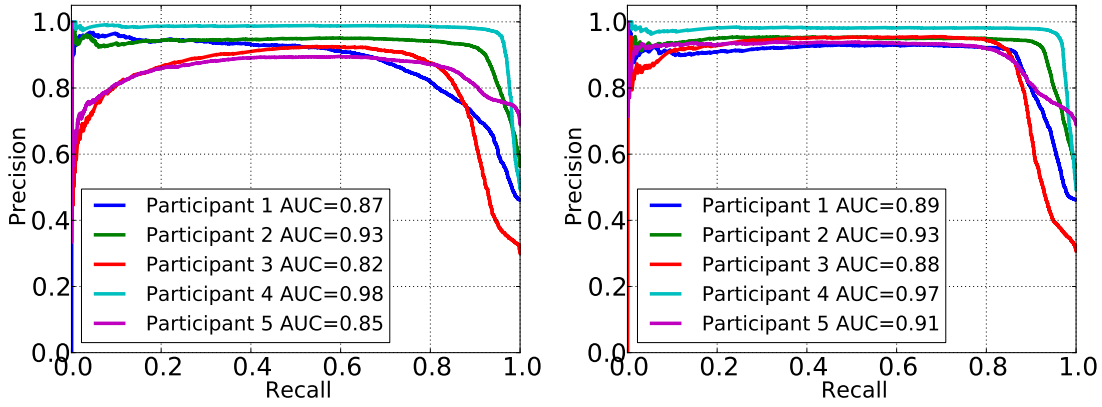
Figure 7.2: Head pose and gaze tracking in a natural dyadic interaction. (a) Original RGB frames of the interviewer (left) and interviewee (right) (b) 3D rendering of the composed 3D scene from two viewpoints, including the estimated head pose and 3D gaze direction. Notice that the camera coordinate system is also rendered.

out moments in which there were facial expressions, blinks, fast head movements, etc.

As manual annotations of gaze are difficult and time-consuming, the annotations consists of only five minutes per interaction. In order to ensure that the timings were accurate, we generated subtitle files and played back the video with the subtitles in VLC, and further adjusted the timing, such that they were accurate.

7.3.2 Alignment and tracking

To obtain the person specific eye image alignment, we used the procedure described in Section 7.2.4. That is, we collected only 3-5 frames in which the participant was gazing at the other subject. Then, assuming the EYEDIAP training set has already been aligned, we computed the test subject's eye alignment parameters using the SDPIA method, as described for the "*eye image alignment for a test subject*" case, in Section 5.4.2. Figure 7.2 shows qualitative results on the obtained head pose and gaze tracking per participant.



(a) (b)

Figure 7.3: Automatic gaze coding precision-recall curves. (a) Without using alignment. (b) Using an eye alignment obtained from the synchronized delaunay implicit parametric alignment approach (cf., Section 7.2.4).

Table 7.1: F_1-score for frame level automatic gaze coding.

	Participant					
Method	1	2	3	4	5	Mean
Head pose	62.5	70.9	49.0	79.4	62.3	64.8
Not aligned	80.9	91.2	83.9	95.7	85.3	87.4
Aligned	87.4	92.3	86.7	96.0	86.9	89.9

7.3.3 Gaze coding results

In Figure 7.3 we show the precision-recall curves obtained by varying τ, from which we can observe the improvement given by the alignment approach. From these curves we obtained the F_1 scores shown in Table 7.1, obtained at Equal-Error Rate τ value. Notice that participants 1, 2, 3 and 5 correspond to interviewees, whereas participant 4 is an example of the interviewer.

In Table 7.1 we also present the results obtained when using only the head pose to code gaze. The approach we used is equivalent to a gaze estimation method based on the head pose direction, as described in Section 5.3.5. However, instead of assuming that the gaze direction is frontal in the **HCS** ($\mathbf{v^h} = [0,0,1]^\top$), we set $\mathbf{v^h}$ to a constant value, obtained as the average gaze reference vector computed over a set of frames in which it is known that the person of interest is looking at the other one (manually annotated). In this manner, we emphasize that a given head pose is more likely when looking at the other subject (for that particular person and interview). In practice, we observed this approach indeed leads to more accurate gaze coding than when based on the default parameters.

The results shown in this table confirm the validity of the proposed approach. The large gap in performance between the head pose based gaze coding, and the methods based on

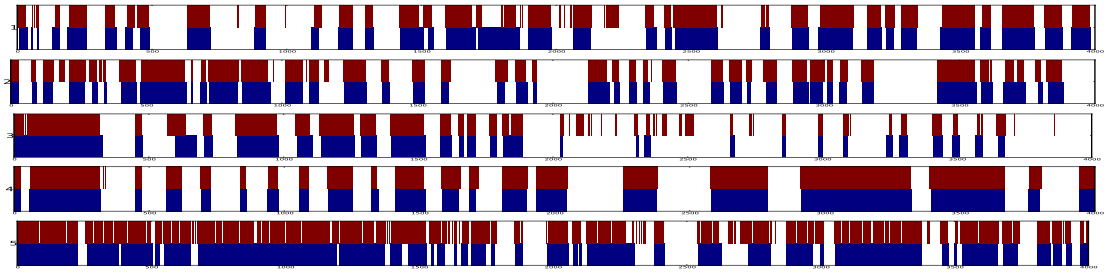


Figure 7.4: Automatic gaze coding results for a sequence of ≈2 minutes. Blue: ground truth. Red: Estimated gaze coding. From top to bottom: participants 1 to 5.

gaze estimation, demonstrate the significant impact and benefit of the head pose and person invariant gaze estimation model we developed in this thesis.

Notice that a person invariant gaze estimation model is required as, in this context, it is not possible to ask the participant to undergo a gaze calibration session. The head pose invariance, on the other hand, was needed as we observed a large diversity in the dynamics of the interactions. The participants alternate between being static and paying attention to the other subject (e.g., when listening), orienting their head in diverse directions (while looking at the other person or not), or even moving their body forward and backward.

It is interesting to observe the behavior of participant 4 in particular (interviewer). As the interviewer is asking questions, with shared attention between either the questionnaire at the table and the interviewee (see Figure 7.2a, left), the head pose is more informative of the gaze behavior (F_1-score is 79.4, in contrast to the average of 64.8). Interviewees in average are more constantly oriented towards the interviewer and are not looking at the interviewer through simple gaze away gestures. Therefore, the head pose is less discriminative of gaze behavior, and the sensing of gaze itself has a more significant impact on the gaze coding accuracy.

In Table 7.1 we also compare the results on gaze coding accuracy when using or not the alignment method we proposed in Chapter 5. Notice that in average the classification accuracy increases when the eye alignment is used. Even though the increase applies to all subjects, there are particular cases (e.g., Participant 1) where an important alignment correction was indeed necessary, and the proposed method was able to find it from very few annotated samples, leading to an significant increase of gaze coding accuracy. This can be futher observed in Figure 7.3, which shows the precision recall obtained when using or not an eye alignment procedure.

Finally, Figure 7.4 shows qualitatively the resulting coding at frame level as a time sequence, whereas Figure 7.5 illustrates a few frames from the interactions which were correctly annotated. Notice that, despite the subtle gaze behavior, the estimates follow closely the ground truth.

Figure 7.5: Qualitative gaze coding results obtained for dyadic interactions in the SONVB database. Per participant, we display the obtained gaze coding class above. Gaze aversion was further decomposed into background classes (up, down, left, right), explained in Section 7.4.2. For all the shown examples, the correct class labels were obtained automatically.

Figure 7.6: The KTH-Idiap corpus recording setup

7.4 Multi-party interactions and method extensions

We recently extended this work to the multi-party situation. Motivated by the need of the research community for corpora which allows the study of group dynamics, we collected the KTH-Idiap Group-Interviewing corpus [Oertel et al., 2014].

We aimed at creating a corpus which encompasses as many different group dynamics on as many different levels as possible. To this end we proposed to study and record group interviews in which several participants are jointly interviewed to get a grant, and which can be seen as an extension of job interviews from the dyadic to the multi-party case. Furthermore, the interview process is composed of different stages which elicit diverse participant's behavior: from self disclosure, to competitive or cooperative interactions. For more details on the scenario, please refer to [Oertel et al., 2014]

This corpus is of high relevance for the problem of automatic gaze coding. The recording setup can be seen in Figure 7.6. We can observe that, in this case, there are four Microsoft Kinect™sensors, but there is still one sensor assigned per participant. Therefore, this is indeed a direct extension of the dyadic setup described in Section 7.2.

7.4.1 Gaze coding

In this section we discuss the elements which need to be extended from the dyadic case, mainly the setup calibration and the gaze coding approach.

Setup calibration

The setup calibration could be addressed in a similar manner as in Section 7.2, i.e., a plane can be fitted to depth measurements of the background walls, from which we can infer the pitch and yaw angles of the **CCS** of each RGB-D sensor. Their translations could then be obtained assuming the room geometry is known.

However, in this corpus, the pose of the sensors is defined a priori by design, as they have the fixed placement shown in Figure 7.6. We only needed to correct for the sensors pitch, which was obtained either automatically, from the background wall's depth data, or manually, whenever the background wall was at a distance exceeding the camera's depth sensing limit.

Once the relative pose between all four **CCS** is obtained, we define a **WCS** to which it is possible to refer all measured quantities.

Multi-party gaze coding

The automatic gaze coding in a multi-party setting follows the same principle as in the dyadic case, described in Section 7.2.3. However we need to account for multiple visual targets. Similarly to the dyadic scenario, we define, for the participant i, a gaze reference vector $\hat{\mathbf{v}}_i^k$ for looking at each other person k. Then, provided that the gaze estimation output is $\mathbf{v}^{\mathbf{h}}$, we define the gaze angle from the participant k as:

$$\psi_i^k = \arccos\left(\mathbf{v}_i^{\mathbf{h}} \cdot \hat{\mathbf{v}}_i^k\right) \tag{7.4}$$

We then obtain the gaze target T_i for the participant i as:

$$T_i = \begin{cases} c & \text{if } \psi_i^c < \tau \\ -1 & \text{otherwise} \end{cases}, \tag{7.5}$$

where $c = \arg\min_k\{\psi_i^k\}$, i.e., the closest participant in terms of gaze, and, "-1" is the *background* class, which indicates that the participant is not looking at any of the targets.

Even though this discussion was made in terms of "participants", the same analysis could be conducted for any type of visual targets, as long as their 3D position can be determined, and therefore, so we can define their corresponding gaze reference vectors.

7.4.2 Background classes

So far we have discriminated between looking at the participant(s) or looking at the background. Nevertheless, it can more informative to further split the background class into sub-classes such as: “up”, “down”, “right”, “left” and “mid-targets”. Andrist et al. [2014] shown that different gaze aways directions could be more characteristic of different functions (turn taking, management, cognitive load, intimacy, etc).

The proposed sub-classes are discriminated by monitoring the gaze pitch and yaw angles, obtained by first referring $\mathbf{v}^{\mathbf{h}}$ to the **WCS** (such that the classes like “up” or “down” are consistent with the **WCS**), and taking into account the positions of the targets. For example, “left” and “right” occur whenever the participant is looking further left or right than the position of *all* the targets in his/her field of view.

7.5 Conclusion

In this Chapter we investigated and proposed solutions for the problem of automatic gaze coding in natural dyadic and multi-party interactions.

The proposed system relies on the 3D tracking of the head pose and gaze direction of the participants, collected using a setup of multiple cameras whose fields of view do not overlap. Then, provided a prior system calibration, the tracked variables can be referred to a unified **WCS** based representation. In this manner, the gaze coding problem reduces to a simple geometric analysis of the 3D head pose and gaze direction of the participants. Furthermore, this approach generalizes well from the dyadic to the multy-party setting. To facilitate the setting up process, we proposed a method to obtain the overall system calibration, which relies on a small quantity of assumptions than can be easily met.

We have conducted evaluations on natural dyadic interactions of real job interviews and reported high accuracy in gaze coding. These experiments and context further validated the importance of the methods we proposed in the previous chapters. Notice that a head pose and person invariant gaze estimation method was indeed necessary to address the problem and context presented in this Chapter.

The obtained results also demonstrate the advantage of using a gaze estimation algorithm over the assumption that the head pose is sufficient to retrieve a subject’s gaze behavior. We also evaluated and confirmed the increase in gaze coding accuracy when using a person specific eye image alignment, inferred from a small set of gaze annotated samples. To this end, we proposed a simple procedure to obtain these gaze annotated samples from a high level verification (is participant A looking at participant B?) which can be easily annotated in a few frames by an experimenter. This further validates the contributions from Chapter 5, and provides a clear example of an application which profit from the proposed eye image alignment and gaze estimation methodology.

Finally, we discussed the extension of the proposed methodology to the multy-party setting, for which we described the handling of multiple subjects and the definition of further background sub-classes which can help to better characterize the gaze patterns of the participants.

We believe the proposed approach can be beneficial for further research on human-human, human-robot and human-computer interaction. For future work it will be interesting to quantitatively evaluate the gaze coding performance in the multi-party scenario. Moreover, in a multi-modal setting, we may leverage conversational priors on human attention, i.e., according to the dynamics of the interaction, at which moments is a participant more likely to gaze at another. Then, provided these soft-annotations, we can automatically extract the few samples needed to infer the person specific eye image alignment. An even more interesting research direction would be to rely on the same soft-annotations and the estimated head positions and gaze directions to jointly solve for the gaze coding and the calibration of the relative pose of each camera. In this manner, the gaze coding system would be fully automatic.

8 Conclusions

8.1 Conclusions

In this thesis we investigated the problem of automatic gaze estimation. We proposed and validated innovative solutions which overcome many of the limitations of previous methods. Thanks to these improvements, the developed methods are suitable for applications that go beyond screen gazing (e.g. HRI, or human behavior analysis), and involving diverse setup configurations and different levels of user cooperation.

We proposed to build a non-intrusive gaze estimation system by relying on remote consumer RGB-D sensors. In this context, we addressed the challenges of low resolution eye image sensing and eye image appearance variations due to head pose. Then, towards minimizing the amount of required user cooperation, we proposed a model-based head pose tracking system which adapts to the current user in order to achieve high accuracy. The proposed head pose tracker does not require any explicit actions from the participant. Similarly, we addressed the problem of eye image appearance variations across subjects, by learning gaze estimation models trained to be person invariant. Finally, we validated the thesis contributions by building upon the proposed methods to address the automatic gaze coding problem in natural dyadic and group interactions.

In the following, we will recall in better detail the thesis contributions and how we addressed the aforementioned challenges. Then, in the following section, we will discuss the limitations and perspectives of the proposed algorithms.

Head pose tracking

We developed a head pose tracker based on the frame by frame registration of a 3D face model to depth data, by using the iterative closest points algorithm. The tracker achives high accuracy, is robust to noise and missing data, and interestingly, defines a head coordinate system in which a stable position of the eye can be defined. Therefore, this method implicitly solves the 3D eye tracking problem, which is required for further gaze sensing related processing stages.

To learn the person specific face model, which is required for the head pose tracking, we proposed to fit a 3D Morphable Model (3DMM) of identity related facial shape variations directly into RGB-D data samples. This can be done in an offline phase, as done by an experimenter who collects and annotate RGB-D samples with facial landmarks. However, to avoid the need for user cooperation, we proposed an online approach, which combines both the head pose tracking problem with the 3DMM fitting in a unified framework. The resulting head pose tracker provides high accuracy to be used for gaze tracking.

EYEDIAP database

We collected and made publicly available a database for the problem of gaze estimation from remote RGB and RGB-D cameras, which we called EYEDIAP. This database systematically isolates the main variables which have an impact on gaze estimation algorithms, such as the head pose, the person specific appearance, the visual target and the ambient conditions. Furthermore, the data was recorded and is provided in diverse modalities, which allows for experimental flexibily, as required by the user. With this contribution we also aimed at encouraging a more principled and objective evaluation of gaze estimation algorithms.

Appearance based gaze estimation

To address the problem of gaze estimation from low resolution sensing, we built upon the appearance based gaze estimation paradigm. To this end, we addressed the following elements:

- *Head pose invariance*. We proposed to use depth measurements as support to rectify the eye region appearance into a canonical head viewpoint, using the estimated head pose. We demonstrated this was a powerful strategy to significantly alleviate the eye image appearance variations due to head pose. This step directly brings head pose invariance to any prior appearance based method developed for a single head pose or head mounted sensors. Alternatively, we can use the 3D face model for the rectification step, where the model's surface is used as a substitute for depth.
- *Eyes coupling*. We proposed a method, based on a sparse reconstruction framework, which includes anatomical constraints on the gaze estimation problem. This slightly improves over the case in which the gaze is estimated separately. Moreover, we found that the strategies based on sparse reconstruction are adequate for situations in which a calibration phase is possible, as it achieves the best accuracy when training from fewer samples.
- *Person invariance*. We addressed the person invariance problem in two manners. In the first one, we proposed to use the sparse reconstruction framework to pre-select subject specific gaze appearance models from a database of subjects. This method resulted in gaze estimation accuracy comparable to using the dataset of all subjects, but at a much reduced computational load. In the second one, we empirically investigated the capability of diverse appearance based methods to train person invariant models. To this end, extensive experiments were conducted using the EYEDIAP database.
- *Eye image alignment*. We raised awareness on the *inter-person eye alignment* problem, which affects the person invariance of appearance based methods. We thus proposed an

alignment algorithm which jointly registers a set of gaze synchronized samples of eye images. This approach is in contrast with previous methods, which use facial features such as eye corners to align the eye images. We empirically validated that this approach improves the performance of person invariant gaze estimation models.

Geometric generative gaze estimation

To address the generalization problems of appearance based methods, which are mostly due to the lack of an eye specific model, we proposed the *geometric generative gaze estimation* approach (G^3E). This is a new paradigm to the gaze estimation problem. It is based on a geometric modelling of the eyeball, gaze activity, and image formation process.

We demonstrated that this method has important advantages with respect to appearance based methods or geometric based methods alone. This method does not require to track local eye features, which makes it adequate for low resolution sensing conditions. This is an important advantage with respect to classical geometric based approaches. Futhermore, as opposed to appearance based methods, we empirically validated that this method was suitable for training from fewer samples, is capable of gaze extrapolation, and has the potential to adapt to different ambient conditions without requiring recalibration.

Automatic gaze coding

We addressed the problem of automatic gaze coding in natural dyadic and group interactions. To this end, we relied on the proposed head pose and person invariant gaze estimation methodology. The system therefore profits from the estimation of these variables in the 3D space to code gaze based on a geometric analysis, which is simple and generalizes to different configurations. We obtained high gaze coding accuracy in challenging real-life scenarios.

8.2 Limitations and perspectives

At the end of each chapter we discussed in detail the limitations of the proposed methods and interesting research directions. In this section, we briefly summarize the main points.

Head pose tracker. Although we obtained high accuracy in head pose estimation, we believe this could be further improved, leading to a reduction on the gaze estimation errors. To this end, the main elements to be considered are constraints based on the visual domain. Note that this will become necessary when the subject is at farther distances from the sensor, where the levels of the depth noise are much higher. There are diverse ways in which the visual domain could be used, for example, i) incorporating normal-flow constraints; ii) using semantic facial landmarks detectors or; iii) using appearance registration methods based on AAM, or by creating a user specific appearance model retrieved online using the shape model. Besides the tracking accuracy, the initialization and failure detection could be improved. In addition, the initialization can very much benefit from the random forest based head detector and pose regressor, which requires the training from a large database to improve its recall and robustness.Failure detection can benefit from appearance based evaluations, e.g., using skin color models or any of the aforementioned visual domain based elements.

Appearance based gaze estimation. In general, these approaches can benefit from the training on larger datasets with further variations in terms of people, illumination conditions, distance to the sensor and expressions. This can help to learn more robust person invariant gaze models. Context information can also help to develop methods to infer the eye image alignment parameters without any supervision. The development of more accurate eye corner detection methods may also help to correct for frame by frame misalignments. Finally, an interesting but more challenging problem is to investigate on the adaptation of pretrained model to a given subject and illumination conditions, which may eventually lead to improved accuracy and robustness.

Geometric generative gaze estimation. We believe there is a significant potential on the G^3E paradigm in general. The main problems to address are the automatic color model distributions learning and adaptation, and the development of robust and fast inference schemes. Nevertheless, there are other interesting research directions. We could further minimize the user cooperation by developing unsupervised and weakly supervised inference schemes, which we argue are plausible under this well defined probabilistic geometric model. Also, notice that, even if the G^3E model was defined in the domain of the pose-rectified eye images, it could be possible to make the image likelihood evaluation process directly in the captured image viewpoint. The challenge in this case would be to define a viewpoint dependent parametric segmentation function.

Automatic gaze coding. The main limitation of the current system is that it is not yet fully automatic. In our experiments we used the offline face model learning (which requires manual intervention), however, the online fitting based tracker could be employed in a newer implementation. More importantly, a careful system calibration is crucial, which therefore requires some level of supervision, either while placing the sensors and doing the distance measurements, or after inferring the calibration parameters, and ensuring the system is well calibrated. The estimation of the eye image alignment parameters also requires the manual intervention to annotate -a few- gazing events. Therefore an interesting research direction is to leverage models of human attention based on conversational dynamics to minimize the level of supervision and setup preparation. The main idea is to infer at what moments it is likely for a participant to be looking at another. In this manner we can design fully unsupervised methodologies for the adaptation/learning of gaze models (for which the G^3E is well suited) and the inference of the setup configuration, such as to estimate the pose of the cameras. This would effectly lead to fully automatic gaze coding system.

Depth dependency. Finally, notice that the proposed system is dependent on the depth modality. This is not a significant limitation as RGB-D sensors are becoming cheaper and more widespread. Nevertheless, we believe many of the proposed contributions could be also applied to situations in which only the visual domain is available, e.g., using the template driven eye image pose-rectification. In such case, the main challenges to address would be the retrieval of the person specific face model and the head pose tracking.

A EYEDIAP measures and benchmarks

A.1 Additional performance measures

Here we define additional performance measures which can be used to compare different gaze estimation algorithms using the EYEDIAP database, but were not exploited in this thesis. Therefore, for an index t in the evaluation set $\mathbf{E}$, with estimated gaze direction $(\mathbf{o}_t, \mathbf{v}_t)$ or screen coordinates $\mathbf{s}_t$, we can define the following error measures:

- **3D distance error** ${\epsilon^d}_t$. This error is defined for the 3D gaze estimation tasks. It conveys how close the estimated 3D gaze ray passes by the visual target 3D position $\hat{\mathbf{p}}_t$, as shown in Eq. A.1.

$$ {\epsilon^d}_t = \min_{\nu} \left\| (\mathbf{o}_t + \nu \mathbf{v}_t) - \hat{\mathbf{p}}_t \right\|_2, \tag{A.1} $$

  where the gaze ray has been defined in parametric form using $\nu \in [0, \infty[$ and $\|.\|_2$ defines the euclidean norm in the 3D space, or the vector's magnitude.

- **Angular error** ${\epsilon^\circ}_t$. This is a normalization alternative to ${\epsilon^d}_t$, where we measure the error in terms of directional error, expressed in degrees:

$$ {\epsilon^\circ}_t = \arcsin\left(\frac{{\epsilon^d}_t}{\left\| \hat{\mathbf{p}}_t - \mathbf{o}_t \right\|_2} \right) \tag{A.2} $$

  However, this is equivalent to the formulation described in Equation A.2.

- **Screen pixel error** ${\epsilon^{\mathbf{s}}}_t$. This error is defined for the gazed screen pixel coordinates prediction task and it is given in Eq. A.3.

$$ {\epsilon^{\mathbf{s}}}_t = \|\mathbf{s}_t - \hat{\mathbf{s}}_t\|_2 \tag{A.3} $$

- **Sensitivity**. When reporting results on a benchmark which evaluates the impact of a given variable on the gaze estimation accuracy (e.g., head pose), it can be useful to compute a normalized measure of the algorithm's robustness to this variable. Assuming the errors

obtained for the baseline conditions are $\epsilon^{\circ 1}$, whereas $\epsilon^{\circ 2}$ corresponds to the errors obtained on the more challenging conditions influenced by the variable under study, the *sensitivity* measure is defined as:

$$\mathscr{R}(\epsilon^{\circ 2}|\epsilon^{\circ 1}) = \max\left(0, \frac{\epsilon^{\circ 2} - \epsilon^{\circ 1}}{\epsilon^{\circ 1}}\right), \tag{A.4}$$

where it is expected that the second experiment is more difficult than the first one (thus normally $\epsilon^{\circ}{}_2 > \epsilon^{\circ}{}_1$ and $\mathscr{R} > 0$). $\mathscr{R}$ is thus intended to measure the robustness of an algorithm with respect to a given variable. For an ideal gaze estimation algorithm, $\mathscr{R} \to 0$. This measure could eventually be reported for Benchmarks 2 and 3, as defined in Section 4.5.

A.2 Additional benchmark

Here we describe a last benchmark which we did not exploit in this thesis, but can be useful in future work.

A.2.1 Benchmark 4: Ambient conditions invariance

In this benchmark the goal is to study the generalization of a gaze estimation algorithm $\mathscr{H}$ to different ambient conditions. To this end, four experiments are conducted for participants 12, 13 and 14, which are the subjects for which some recording sessions were repeating involving different sensing (ambient) conditions. Notice that only the floating target (*FT*) data is available for this task. For each participant k, the experiments are:

- **Experiment 1**. Let **T** be the first half of session k-A-FT-S and **D** be the second half of session k-A-FT-S. The obtained mean angular gaze estimation error is $\epsilon^{\circ A}$.
- **Experiment 2**. The first half of session k-A-FT-S is again used as training set **T**, but now the test condition is changed by using the second half of session k-B-FT-S as test set **D**. The obtained mean angular error is $\epsilon^{\circ B|A}$.
- **Experiment 3**. We conduct similar experiments, but now in the reverse order. That is, the first half of session k-B-FT-S is assigned to the training set **T**, and the second half of session k-B-FT-S is assigned to the test set **D**. The obtained mean angular error is $\epsilon^{\circ B}$.
- **Experiment 4**. The first half of session k-B-FT-S is assigned to **T** and the A-condition data is used for testing, i.e. the second temporal half of session k-A-FT-S is used as test data **D**. The obtained mean angular error is $\epsilon^{\circ A|B}$.

For each participant $\epsilon^{\circ A}$, $\epsilon^{\circ B|A}$, $\epsilon^{\circ B}$ and $\epsilon^{\circ A|B}$ are computed, together with $\epsilon^{\circ} = (\epsilon^{\circ A} + \epsilon^{\circ B})/2$ and $\mathscr{R} = (\mathscr{R}(\epsilon^{\circ B|A}|\epsilon^{\circ A}) + \mathscr{R}(\epsilon^{\circ A|B}|\epsilon^{\circ B}))/2$. As final result the average of ϵ° and $\mathscr{R}$, among the 3 participants, is reported.

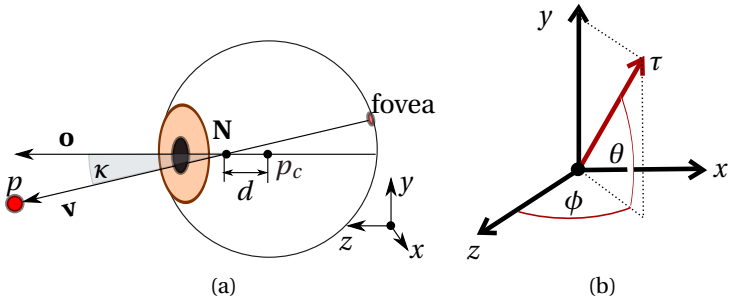
B G^3E derivations

B.1 Eye geometric model

Fig B.1 depicts the used geometric model. This is a standard representation of the human eyeball geometry. If all geometric parameters are known then, from the position of p, we can estimate the necessary eyeball orientation, characterized by the optical axis (**o**), such that **v** intersects the visual target. This process is denoted as follows:

$$\mathbf{o}(p) = (f_\phi(p;\kappa,d,p_c), f_\theta(p;\kappa,d,p_c)), \tag{B.1}$$

for which we need to find the functions f_ϕ and f_θ. First we define the transformations between the vectorial and angular representation of an "axis" where $\tau = (\phi_\tau, \theta_\tau)$ represents the angles of the axis (as in Fig. B.1b) and $\tau_v \in \mathbb{R}^3$ is the equivalent 3D vector:



(a) (b)

Figure B.1: (a) Eye geometry with optical (**o**) and visual (**v**) axis definition. (b) spherical parametrization of an axis "τ".

$$\tau_v = \Psi(\tau) = \begin{pmatrix} \cos(\theta_\tau)\sin(\phi_\tau) \\ \sin(\theta_\tau) \\ \cos(\theta_\tau)\cos(\phi_\tau) \end{pmatrix}$$

The inverse transformation (assuming τ_v is a unit vector) is given by

$$\tau = \Psi^{-1}(\tau_v) = \begin{pmatrix} \tan^{-1}(\tau_{vx}/\tau_{vz}) \\ \sin^{-1}(\tau_{vy}) \end{pmatrix}$$

Then the problem consist in finding the axis orientation $\mathbf{o}$ (or eye rotation) such that the following equation is satisfied:

$$p = p_c + d\Psi(\mathbf{o}) + d_p\Psi(\mathbf{o}+\kappa)$$

Where d_p is the distance between $\mathbf{N}$ and p. It can then be shown that the necessary eye rotation is:

$$f_\phi(p;\kappa,d,p_c) = \tan^{-1}\left(\frac{p_x - p_{cx}}{p_z - p_{cz}}\right) - \tan^{-1}\left(\frac{d_p\cos(\theta_\kappa)\sin(\phi_\kappa)}{d + d_p\cos(\theta_\kappa)\cos(\phi_\kappa)}\right) \tag{B.2}$$

and:

$$f_\theta(p;\kappa,d,p_c) = \sin^{-1}\left(\frac{p_y - p_{cy}}{|p - p_c|}\right) - \sin^{-1}\left(\frac{d_p\sin(\theta_\kappa)}{|d\mathbf{z} + d_p\Psi(\kappa)|}\right) \tag{B.3}$$

For which $\mathbf{z} := [0,0,1]^\top$ and:

$$d_p = \sqrt{d^2\cos^2(\theta_\kappa)\cos^2(\phi_\kappa) - d^2 + |p - p_c|^2} - d\cos(\theta_\kappa)\cos(\phi_\kappa) \tag{B.4}$$

B.2 Segmentation function

Here we briefly describe the parametric segmentation derived for the G^{3}E model. It is composed of two main elements: the cornea-sclera segmentation and the skin segmentation.

B.2.1 Cornea-sclera segmentation

Given the parameters r_c, r_e, p_c and $\mathbf{o}$ we need to determine the contour of the cornea which is projected in the x, y plane and then transformed to image coordinates u, v. Using a parametric representation of the circumference of the iris (limbus), where $t \in [-\pi, \pi[$, we obtain:

$$\hat{\mathbf{x}}(t) = \begin{bmatrix} r_c \cos(t) \\ r_c \sin(t) \\ \sqrt{r_e^2 - r_c^2} \end{bmatrix}$$

This is valid in the coordinate system located at the eyeball center p_c. If we apply a rotation of the eyeball given by $\mathbf{o} = (\phi, \theta)$, then the parametric contour is given by:

$$\hat{\mathbf{x}}(t;\mathbf{o}) = \begin{bmatrix} \cos(\phi) & -\sin(\theta)\sin(\phi) & \sin(\phi)\cos(\theta) \\ 0 & \cos(\theta) & \sin(\theta) \\ -\sin(\phi) & -\sin(\theta)\cos(\phi) & \cos(\phi)\cos(\theta) \end{bmatrix} \begin{bmatrix} r_c \cos(t) \\ r_c \sin(t) \\ \sqrt{r_e^2 - r_c^2} \end{bmatrix}$$

Expanding and now including the center of the eyeball p_c as a translation:

$$\hat{\mathbf{x}}(t;\mathbf{o},p_c) = \begin{bmatrix} r_c \cos(\phi)\cos(t) - r_c \sin(\theta)\sin(\phi)\sin(t) + \sin(\phi)\cos(\theta)\sqrt{r_e^2 - r_c^2} \\ r_c \cos(\theta)\sin(t) + \sqrt{r_e^2 - r_c^2}\sin(\theta) \\ -r_c \sin(\phi)\cos(t) - r_c \sin(\theta)\cos(\phi)\sin(t) + \cos(\phi)\cos(\theta)\sqrt{r_e^2 - r_c^2} \end{bmatrix} + p_c$$

As we are mostly interested in the projection in the x, y plane, the final contour is given as:

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} r_c \cos(\phi)\cos(t) - r_c \sin(\theta)\sin(\phi)\sin(t) + \sin(\phi)\cos(\theta)\sqrt{r_e^2 - r_c^2} + p_{cx} \\ r_c \cos(\theta)\sin(t) + \sqrt{r_e^2 - r_c^2}\sin(\theta) + p_{cy} \end{bmatrix} \tag{B.5}$$

Transformation to image coordinates is straightforward: the projection to the pose-rectified eye images is orthogonal and the pixel size is given (due to the available depth data and camera calibration).

B.2.2 Eyelids segmentation

The upper and lower eyelids are defined as quadratic Bezier curves. For example, the upper eyelid has as control points k_l, $\mathbf{c}_u$ and k_r, i.e. the eye corners and the middle point $\mathbf{c}_u$. We defined $\mathbf{c}_u := [(k_{lu} + k_{ru})/2, u_e]$, where u_e define the vertical position of the control point and thus, the eyelid opening. The lower eyelid is defined in the same manner, but using l_e to define the eyelid opening (vertical).

B.2.3 Final segmentation

A segmentation is therefore defined by a set of geometric parameters: r_c, r_e, p_c, k_l, k_r, u_e, l_e and $\mathbf{o}$. Using Eq. B.5 we can query if a pixel u, v is inside the contour by simple comparisons. Similarly we can query if the pixel u, v is or not within the skin region by comparisons to the eyelids Bezier curves. Notice the skin class overrides the sclera and cornea classes, as the skin occludes these regions. This set of rules thus define the segmentation function $S_{u,v}$.

B.3 Variational Bayes

In this section we summarize the main generalities of Variational Bayes which are relevant to the approach described in chapter 6[1]. In VB the posterior $p(\mathbf{Z}|\mathbf{X})$ is approximated a the *proposal distribution* $q(\mathbf{Z})$. This approximation leads to the known relation:

$$\ln p(\mathbf{X}) = \mathscr{L}(q) + \mathrm{KL}(q||p) \tag{B.6}$$

Where $\mathrm{KL}(q||p)$ is the Kullback-Leibler (KL) divergence between $p(\mathbf{Z}|\mathbf{X})$ and $q(\mathbf{Z})$, and $\ln p(\mathbf{X})$ is the log-marginal likelihood, which is a constant quantity under a fix model. In general, the *variational lower bound* $\mathscr{L}(q)$ takes the following form:

$$\begin{aligned}\mathscr{L}(q) &= \int q(\mathbf{Z}) \ln p(\mathbf{X}, \mathbf{Z}) d\mathbf{Z} - \int \ln q(\mathbf{Z}) q(\mathbf{Z}) d\mathbf{Z} \\ &= \mathbb{E}_{\mathbf{Z}}\left[\ln p(\mathbf{X}, \mathbf{Z})\right] - \mathbb{E}_{\mathbf{Z}}\left[\ln q(\mathbf{Z})\right]\end{aligned} \tag{B.7}$$

Where $p(\mathbf{X}, \mathbf{Z})$ is the complete joint distribution and expectations are defined with respect to $q(\mathbf{Z})$. The goal is to maximize the functional $\mathscr{L}$ with respect to $q(\mathbf{Z})$ as, due to $\ln p(\mathbf{X})$ being constant, this is equivalent to minimize the KL divergence between $p(\mathbf{Z}|\mathbf{X})$ and $q(\mathbf{Z})$. Therefore the optimal $q(\mathbf{Z})$ becomes a tractable substitute for the posterior.

[1] For further information on Variational Bayes (VB), please refer to Bishop [2007]

Assuming we have a training set $\{(\mathbf{I}^i, p^i)\}_{i=1}^N$ of N pairs of eye images $\mathbf{I}$ and visual targets p, indexed by i, the functional form of $q(\mathbf{Z})$ is given by:

$$\begin{aligned} q(\mathbf{Z}) = &\mathcal{N}(\mu_d, \sigma_d)\mathcal{N}(\mu_{\phi_\kappa}, \sigma_{\phi_\kappa})\mathcal{N}(\mu_{\theta_\kappa}, \sigma_{\theta_\kappa})\mathcal{N}(\mu_{r_e}, \sigma_{r_e})\mathcal{N}(\mu_{r_c}, \sigma_{r_c})\mathcal{N}(\mu_{p_{cx}}, \sigma_{p_{cx}}) \\ &\mathcal{N}(\mu_{p_{cy}}, \sigma_{p_{cy}})\mathcal{N}(\mu_{p_{cz}}, \sigma_{p_{cz}})\mathcal{N}(\mu_{k_{lu}}, \sigma_{k_{lu}})\mathcal{N}(\mu_{k_{lv}}, \sigma_{k_{lv}})\mathcal{N}(\mu_{k_{ru}}, \sigma_{k_{ru}})\mathcal{N}(\mu_{k_{rv}}, \sigma_{k_{rv}}) \\ &\prod_{i=1}^{N}\left[\mathcal{N}(\mu_\phi^i, \sigma_\phi^i)\mathcal{N}(\mu_\theta^i, \sigma_\theta^i)\mathcal{N}(\mu_{u_e}^i, \sigma_{u_e}^i)\mathcal{N}(\mu_{l_e}^i, \sigma_{l_e}^i)\prod_{u,v\in i} q(\lambda_i^{u,v})\right] \end{aligned} \tag{B.8}$$

Where $\mathcal{N}(\mu_a, \sigma_a)$ is a simplification for the univariate normal distribution $\mathcal{N}(a|\mu_a, \sigma_a)$. Notice that $q(\mathbf{Z})$ can be seen as a fully factorized distribution where $q(\mathbf{Z}) = \prod_j q(\mathbf{Z}_j)$. If we allow variations of $\mathcal{L}$ around a single factor $q_j = q(\mathbf{Z}_j)$ then Eq. B.7 is equivalent to the following expression:

$$\mathcal{L}(q) = \mathbb{E}_{\mathbf{Z}_j}\left[\mathbb{E}_{\mathbf{Z}_{i\neq j}}\left[\ln p_j(\mathbf{X}, \mathbf{Z})\right]\right] - \mathbb{E}_{\mathbf{Z}_j}\left[\ln q_j(\mathbf{Z}_j)\right] + cst_j \tag{B.9}$$

Where cst_j is a constant with respect to $\mathbf{Z}_j$, and thus constant for any change of q_j. Notice that computations related to q_j in Eq. B.9, ignoring cst_j, need only the additive terms from $\ln p(\mathbf{X}, \mathbf{Z})$ which include the variable $\mathbf{Z}_j$. Here we denote these terms as $\ln p_j(\mathbf{X}, \mathbf{Z})$.

In standard VB, $\mathcal{L}$ can be maximized in an iterative fashion, updating a factor per iteration until global convergence. If $\mathcal{L}$ is optimized with respect to the factor q_j, while keeping the rest of factors unchanged, the optimal distribution for the factor $\mathbf{Z}_j$ can be shown to be:

$$\ln q_j^*(\mathbf{Z}_j) = \mathbb{E}_{\mathbf{Z}_{i\neq j}}\left[\ln p_j(\mathbf{Z}, \mathbf{X})\right] + cst_j \tag{B.10}$$

Depending on the definition of the model conditionals and priors Eq. B.10 normally can be solved analytically. This is not the case for the continuous variables of our model due to the complex relations in f_ϕ, f_θ and $S_{u,v}$.To solve this aspect we imposed a parametric form for the continuous factors: a univariate Gaussian. This leads to an optimization of $\mathcal{L}$ with respect to the Gaussian parameters.

Given the -univariate- Gaussian factor $q(\mathbf{Z}_j) = \mathcal{N}(\mu_{\mathbf{Z}_j}, \sigma_{\mathbf{Z}_j})$ we followed the approach described by Opper and Archambeau [2009], where the derivatives can be computed as shown

in Eq. B.11 and Eq. B.12.

$$\frac{\partial \mathscr{L}}{\partial \mu_{\mathbf{Z}_j}} = \mathbb{E}_{\mathbf{Z}} \left[\frac{\left(\mathbf{Z}_j - \mu_{\mathbf{Z}_j}\right)}{\sigma^2_{\mathbf{Z}_j}} \ln p_j(\mathbf{Z},\mathbf{X}) \right] \tag{B.11}$$

$$\frac{\partial \mathscr{L}}{\partial \sigma_{\mathbf{Z}_j}} = \mathbb{E}_{\mathbf{Z}} \left[\frac{\left(\left(\mathbf{Z}_j - \mu_{\mathbf{Z}_j}\right)^2 - \sigma^2_{\mathbf{Z}_j}\right)}{\sigma^3_{\mathbf{Z}_j}} \ln p_j(\mathbf{Z},\mathbf{X}) \right] + \frac{1}{\sigma_{\mathbf{Z}_j}} \tag{B.12}$$

B.4 Outliers term

To obtain the outliers update we first identify the additive terms of $\ln p(\mathbf{X},\mathbf{Z})$ which include $\lambda_i^{u,v}$ (i.e., the Bernoulli prior for λ and the image likelihood) and letting any other term being absorbed by cst:

$$\ln p_{\lambda_i^{u,v}}(\mathbf{Z},\mathbf{X}) = \lambda_i^{u,v}(\ln\hat{\omega} + \ln\epsilon) + (1-\lambda_i^{u,v})\left(\ln(1-\hat{\omega}) + \sum_l S_{u,v}(l;\mathbf{m},\mathbf{s},p_c)\ln p(c^{u,v}|\Lambda_l)\right) + cst \tag{B.13}$$

The optimal distribution for $\lambda_i^{u,v}$ can be obtained by solving Eq. B.10:

$$\begin{aligned}\ln q^*(\lambda_i^{u,v}) &= \mathbb{E}_{\mathbf{Z}\neq\lambda_i^{u,v}}\left[\lambda_i^{u,v}(\ln\hat{\omega}+\ln\epsilon) + (1-\lambda_i^{u,v})\left(\ln(1-\hat{\omega}) + \sum_l S_{u,v}(l;\mathbf{m},\mathbf{s},p_c)\ln p(c^{u,v}|\Lambda_l)\right)\right] + cst \\ &= \lambda_i^{u,v}(\ln\hat{\omega}+\ln\epsilon) + (1-\lambda_i^{u,v})\left(\ln(1-\hat{\omega}) + \sum_l \mathbb{E}_{\mathbf{m},\mathbf{s},p_c}\left[S_{u,v}(l;\mathbf{m},\mathbf{s},p_c)\right]\ln p(c_{x,y}|\Lambda_l)\right) + cst\end{aligned} \tag{B.14}$$

By inspection it can be shown that Eq. B.14 is equivalent to the following Bernoulli distribution:

$$q^*(\lambda_i^{u,v}) = (\omega_i^{u,v})^{\lambda_i^{u,v}}(1-\omega_i^{u,v})^{(1-\lambda_i^{u,v})} \tag{B.15}$$

Where

$$\omega_i^{u,v} = \frac{\hat{\omega}}{(1-\hat{\omega})\frac{1}{c}\prod_l p(c|\Lambda_l)^{\mathbb{E}_{\mathbf{m},\mathbf{s},p_c}[S_{u,v}(l;\mathbf{m},\mathbf{s},p_c)]} + \hat{\omega}} \tag{B.16}$$

Notice that $\mathbb{E}_{\mathbf{m},\mathbf{s},p_c}\left[S_{u,v}(l;\mathbf{m},\mathbf{s},p_c)\right]$ is an expectation with respect to $q(\mathbf{m}_i)q(\mathbf{s})q(p_c)$, which we can compute using a Monte Carlo sampling approximation.

B.5 Gaussian derivatives

In this section we list the derivatives for our model, obtained by solving Eq. B.11 and Eq. B.12 for each parameter.

B.5.1 Common expressions

We will find useful to define expressions which are derived from log of the conditionals related to the visual axis gazing action and image likelihood for a given sample i:

$$g_i(\mathbf{a},p_c) := -\frac{1}{2\hat{\sigma}_{\mathbf{o}}^2}\left(\left(f_\phi(p^i;p_c,d,\kappa)-\mu_\phi^i\right)^2+\left(f_\theta(p^i;p_c,d,\kappa)-\mu_\theta^i\right)^2\right) \tag{B.17}$$

$$h_i(\mathbf{m},\mathbf{s},p_c) := \sum_{u,v}(1-\omega_i^{u,v})\sum_{l=1}^{3} S_{u,v}(l;\mathbf{m},\mathbf{s},p_c)\ln p(c_i^{u,v}|\Lambda_l) \tag{B.18}$$

For a random variable a with a prior Gaussian distribution $\mathcal{N}(\hat{\mu}_a,\hat{\sigma}_a)$ and an associated proposal distribution $q(a)=\mathcal{N}(\mu_a,\sigma_a)$ it can be shown that:

$$\mathbb{E}_a\left[\ln\mathcal{N}(\hat{\mu}_a,\hat{\sigma}_a)\right] = -\frac{(\mu_a^2+\sigma_a^2-2\mu_a\hat{\mu}_a)}{2\hat{\sigma}_a^2}+cst \tag{B.19}$$

$$\mathbb{E}_a\left[\frac{(a-\mu_a)}{\sigma_a^2}\ln\mathcal{N}(\hat{\mu}_a,\hat{\sigma}_a)\right] = -\frac{(\mu_a-\hat{\mu}_a)}{\hat{\sigma}_a^2} \tag{B.20}$$

$$\mathbb{E}_a\left[\frac{\left((a-\mu_a)^2-\sigma_a^2\right)}{\sigma_a^3}\ln\mathcal{N}(\hat{\mu}_a,\hat{\sigma}_a)\right]=-\frac{\sigma_a}{\hat{\sigma}_a^2} \tag{B.21}$$

For the special case of the term which correlates the eyelid opening with the eye elevation:

$$\mathbb{E}_{\theta^i,u_e^i}\left[\ln\mathcal{N}(a_u\theta^i+b_u,\hat{\sigma}_{u_e})\right]=-\frac{{\mu_{u_e}^i}^2+{\sigma_{u_e}^i}^2-2a_u\mu_{u_e}^i\mu_\theta^i-2b_u\mu_{u_e}^i+a_u^2{\mu_\theta^i}^2+a_u^2{\sigma_\theta^i}^2+2a_ub_u\mu_\theta^i+b_u^2}{2\hat{\sigma}_{u_e}^2} \tag{B.22}$$

As described in Chapter 6, all parameters are divided in sub-groups. The overall optimization is done by iteratively optimizing over these sub-groups until global convergence. In the following we describe each of the groups with their corresponding bound and set of derivatives for the optimization. In each case, a gradient ascent method is used to find the optimal values.

B.5.2 Eye corners and eyelids opening

This group is optimized for the eyelids corners and the eyelids opening for all samples jointly, such that we optimize for the group of random variables: $\mathbf{e}:=(k_{lu},k_{lv},k_{ru},k_{rv},{l_e}^1,{u_e}^1,\ldots,{l_e}^N,{u_e}^N)$.

To this end we identify the additives terms of $\ln p(\mathbf{X},\mathbf{Z})$ which include $\mathbf{e}$, as in Eq. B.9. In this way we obtain the variational lower bound in function of only the terms related to $\mathbf{e}$ as follows:

$$\begin{aligned}\mathcal{L}(q)\approx&\mathbb{E}_{k_{lu}}\left[\ln\mathcal{N}(\hat{\mu}_{k_{lu}},\hat{\sigma}_{k_{lu}})\right]+\mathbb{E}_{k_{lv}}\left[\ln\mathcal{N}(\hat{\mu}_{k_{lv}},\hat{\sigma}_{k_{lv}})\right]+\mathbb{E}_{k_{ru}}\left[\ln\mathcal{N}(\hat{\mu}_{k_{ru}},\hat{\sigma}_{k_{ru}})\right]+\mathbb{E}_{k_{rv}}\left[\ln\mathcal{N}(\hat{\mu}_{k_{rv}},\hat{\sigma}_{k_{rv}})\right]+\\&\ln\sigma_{k_{lu}}+\ln\sigma_{k_{lv}}+\ln\sigma_{k_{ru}}+\ln\sigma_{k_{rv}}+\sum_i\Big[\mathbb{E}_{l_e^i}\left[\ln\mathcal{N}(\hat{\mu}_{l_e},\hat{\sigma}_{l_e})\right]+\mathbb{E}_{\theta^i,u_e^i}\left[\ln\mathcal{N}(a_u\theta^i+b_u,\hat{\sigma}_{u_e})\right]+\\&\ln\sigma_{u_e}^i+\ln\sigma_{l_e}^i+\frac{1}{M}\sum_{k=1}^M h_i(\mathbf{m}^k,\mathbf{s}^k,p_c^k)\Big]+K_{\mathbf{e}}\end{aligned} \tag{B.23}$$

Where $K_{\mathbf{e}}$ is a constant. We can refer to Eq. B.19 to solve the expectations for the terms related to the priors. In the case of the Jacobian, we develop Eq. B.11 and Eq. B.12, and taking into account the results shown in Eq. B.20 and Eq. B.21, we will define the eye corners Jacobian $\mathbf{J_{ek}}$ and the -per sample i- eyelids opening Jacobian as $\mathbf{J}_\mathbf{e}^i$. These terms are computed as shown in

Eq. B.24 and Eq. B.25:

$$\mathbf{J_{ek}} := \begin{bmatrix} \frac{\partial \mathscr{L}}{\partial \mu_{k_{lu}}} \\ \frac{\partial \mathscr{L}}{\partial \sigma_{k_{lu}}} \\ \frac{\partial \mathscr{L}}{\partial \mu_{k_{lv}}} \\ \frac{\partial \mathscr{L}}{\partial \sigma_{k_{lv}}} \\ \frac{\partial \mathscr{L}}{\partial \mu_{k_{ru}}} \\ \frac{\partial \mathscr{L}}{\partial \sigma_{k_{ru}}} \\ \frac{\partial \mathscr{L}}{\partial \mu_{k_{rv}}} \\ \frac{\partial \mathscr{L}}{\partial \sigma_{k_{rv}}} \end{bmatrix} \approx - \begin{bmatrix} \frac{\left(\mu_{k_{lu}} - \hat{\mu}_{k_{lu}}\right)}{\hat{\sigma}^2_{k_{lu}}} \\ \frac{\sigma_{k_{lu}}}{\hat{\sigma}^2_{k_{lu}}} - \frac{1}{\sigma_{k_{lu}}} \\ \frac{\left(\mu_{k_{lv}} - \hat{\mu}_{k_{lv}}\right)}{\hat{\sigma}^2_{k_{lv}}} \\ \frac{\sigma_{k_{lv}}}{\hat{\sigma}^2_{k_{lv}}} - \frac{1}{\sigma_{k_{lv}}} \\ \frac{(\mu_{k_{ru}} - \hat{\mu}_{k_{ru}})}{\hat{\sigma}^2_{k_{ru}}} \\ \frac{\sigma_{k_{ru}}}{\hat{\sigma}^2_{k_{ru}}} - \frac{1}{\sigma_{k_{ru}}} \\ \frac{(\mu_{k_{rv}} - \hat{\mu}_{k_{rv}})}{\hat{\sigma}^2_{k_{rv}}} \\ \frac{\sigma_{k_{rv}}}{\hat{\sigma}^2_{k_{rv}}} - \frac{1}{\sigma_{k_{rv}}} \end{bmatrix} + \frac{1}{M} \sum_{i=1}^{N} \sum_{k=1}^{M} \begin{bmatrix} \frac{\left({k_{lu}}^k - \mu_{k_{lu}}\right)}{\sigma^2_{k_{lu}}} \\ \frac{\left(\left({k_{lu}}^k - \mu_{k_{lu}}\right)^2 - \sigma^2_{k_{lu}}\right)}{{\sigma_{k_{lu}}}^3} \\ \frac{\left({k_{lv}}^k - \mu_{k_{lv}}\right)}{\sigma^2_{k_{lv}}} \\ \frac{\left(\left({k_{lv}}^k - \mu_{k_{lv}}\right)^2 - \sigma^2_{k_{lv}}\right)}{{\sigma_{k_{lv}}}^3} \\ \frac{\left({k_{ru}}^k - \mu_{k_{ru}}\right)}{\sigma^2_{k_{ru}}} \\ \frac{\left(\left({k_{ru}}^k - \mu_{k_{ru}}\right)^2 - \sigma^2_{k_{ru}}\right)}{{\sigma_{k_{ru}}}^3} \\ \frac{\left({k_{rv}}^k - \mu_{k_{rv}}\right)}{\sigma^2_{k_{rv}}} \\ \frac{\left(\left({k_{rv}}^k - \mu_{k_{rv}}\right)^2 - \sigma^2_{k_{rv}}\right)}{{\sigma_{k_{rv}}}^3} \end{bmatrix} h_i(\mathbf{m}^k, \mathbf{s}^k, p_c^k) \tag{B.24}$$

And the Jacobian related to the eyelids opening of a sample i, $\mathbf{J}^i_{\mathbf{e}}$ is given as:

$$\mathbf{J}^i_{\mathbf{e}} := \begin{bmatrix} \frac{\partial \mathscr{L}}{\partial \mu^i_{u_e}} \\ \frac{\partial \mathscr{L}}{\partial \sigma^i_{u_e}} \\ \frac{\partial \mathscr{L}}{\partial \mu^i_{l_e}} \\ \frac{\partial \mathscr{L}}{\partial \sigma^i_{l_e}} \end{bmatrix} \approx - \begin{bmatrix} \frac{\left(\mu^i_{u_e} - (a_u \mu^i_\theta + b_u)\right)}{\hat{\sigma}^2_{u_e}} \\ \frac{\sigma^i_{u_e}}{\hat{\sigma}^2_{u_e}} - \frac{1}{\sigma^i_{u_e}} \\ \frac{\left(\mu^i_{l_e} - \hat{\mu}_{l_e}\right)}{\hat{\sigma}^2_{l_e}} \\ \frac{\sigma^i_{l_e}}{\hat{\sigma}^2_{l_e}} - \frac{1}{\sigma^i_{l_e}} \end{bmatrix} + \frac{1}{M} \sum_{k=1}^{M} \begin{bmatrix} \frac{\left({u_e}^k - \mu^i_{u_e}\right)}{{\sigma^i_{u_e}}^2} \\ \frac{\left(\left({u_e}^k - \mu^i_{u_e}\right)^2 - {\sigma^i_{u_e}}^2\right)}{{\sigma^i_{u_e}}^3} \\ \frac{\left({l_e}^k - \mu^i_{l_e}\right)}{{\sigma^i_{l_e}}^2} \\ \frac{\left(\left({l_e}^k - \mu^i_{l_e}\right)^2 - {\sigma^i_{l_e}}^2\right)}{{\sigma^i_{l_e}}^3} \end{bmatrix} h_i(\mathbf{m}^k, \mathbf{s}^k, p_c^k) \tag{B.25}$$

Which relies on the result in Eq. B.22. Finally we can define the group Jacobian as $\mathbf{J_e} = \left[\mathbf{J_{ek}}^\top, {\mathbf{J}^1_{\mathbf{e}}}^\top, \ldots, {\mathbf{J}^N_{\mathbf{e}}}^\top\right]^\top$, used to optimize the eyelids corners and opening for all samples jointly.

Notice the Monte Carlo expectations are based on drawing M samples $\{(\mathbf{m}^k, \mathbf{s}^k, p_c^k)\}_{k=1}^M$ from the current distribution $q(\mathbf{s})q(p_c)q(\mathbf{m}^i)$ (this is done per sample i) and notice that $\mathbf{m}^k := (\phi^k, \theta^k, u_e^k, l_e^k)$.

B.5.3 Eyeball geometry and orientation

The eyeball geometry and orientation are optimized jointly, i.e. with respect to the random variables $\mathbf{g} := \left(p_{cx}, p_{cy}, r_e, r_c, \phi^1, \theta^1, \ldots, \phi^N, \theta^N\right)$. To this end we define the related variational lower bound by using only the additive terms of $\ln p(\mathbf{X}, \mathbf{Z})$ which include $\mathbf{g}$, leading to:

$$\begin{aligned}\mathscr{L}(q) \approx & \mathbb{E}_{p_{cx}}\left[\ln \mathcal{N}(\hat{\mu}_{p_{cx}}, \hat{\sigma}_{p_{cx}})\right] + \mathbb{E}_{p_{cy}}\left[\ln \mathcal{N}(\hat{\mu}_{p_{cy}}, \hat{\sigma}_{p_{cy}})\right] + \mathbb{E}_{r_e}\left[\ln \mathcal{N}(\hat{\mu}_{r_e}, \hat{\sigma}_{r_e})\right] + \mathbb{E}_{r_c}\left[\ln \mathcal{N}(\hat{\mu}_{r_c}, \hat{\sigma}_{r_c})\right] + \\ & \ln \sigma_{p_{cx}} + \ln \sigma_{p_{cy}} + \ln \sigma_{r_e} + \ln \sigma_{r_c} + \sum_i \left[\mathbb{E}_{\phi^i}\left[\ln \mathcal{N}(\hat{\mu}_{\phi^i}, \hat{\sigma}_{\phi^i})\right] + \mathbb{E}_{\theta^i}\left[\ln \mathcal{N}(\hat{\mu}_{\theta^i}, \hat{\sigma}_{\theta^i})\right] + \right. \\ & \left. \mathbb{E}_{\theta^i, u_e^i}\left[\ln \mathcal{N}(a_u \theta^i + b_u, \hat{\sigma}_{u_e})\right] + \ln \sigma_{\phi^i} + \ln \sigma_{\theta^i} + \frac{1}{M}\sum_{k=1}^{M}\left(g_i(\mathbf{a}^k, p_c^k) + h_i(\mathbf{m}^k, \mathbf{s}^k, p_c^k)\right)\right] + K_{\mathbf{g}}\end{aligned} \tag{B.26}$$

For which we have defined the expected eye orientation conditioned on the position of the visual target p^i:

$$\hat{\mu}_{\phi^i} = \mathbb{E}_{\mathbf{a}, p_c}\left[f_\phi(p^i, \mathbf{a}, p_c)\right] \approx \frac{1}{M}\sum_{k=1}^{M} f_\phi(p^i; \mathbf{a}^k, p_c^k) \tag{B.27}$$

$$\hat{\mu}_{\theta^i} = \mathbb{E}_{\mathbf{a}, p_c}\left[f_\theta(p^i, \mathbf{a}, p_c)\right] \approx \frac{1}{M}\sum_{k=1}^{M} f_\theta(p^i; \mathbf{a}^k, p_c^k) \tag{B.28}$$

Notice that Eq. B.27 and Eq. B.28 are key elements into the well constrained global optimization, as these values put constraints on the eye orientation conditioned by the visual target position, while comparing to image data. These quantities assume drawing a set of samples $\{(\mathbf{a}^k, p_c^k)\}_{k=1}^M$ from $q(\mathbf{a}, p_c)$.

In the case of the eyeball location, in terms of the x and y coordinates, it takes information from both image and target data. We thus define the corresponding Jacobian $\mathbf{J}_{p_{cxy}}$ as follows:

$$\mathbf{J}_{p_{cxy}} := \begin{bmatrix} \frac{\partial \mathscr{L}}{\partial \mu_{p_{cx}}} \\ \frac{\partial \mathscr{L}}{\partial \sigma_{p_{cx}}} \\ \frac{\partial \mathscr{L}}{\partial \mu_{p_{cy}}} \\ \frac{\partial \mathscr{L}}{\partial \sigma_{p_{cy}}} \end{bmatrix} \approx -\begin{bmatrix} \frac{(\mu_{p_{cx}} - \hat{\mu}_{p_{cx}})}{\hat{\sigma}^2_{p_{cx}}} \\ \frac{\sigma_{p_{cx}}}{\hat{\sigma}^2_{p_{cx}}} - \frac{1}{\sigma_{p_{cx}}} \\ \frac{(\mu_{p_{cy}} - \hat{\mu}_{p_{cy}})}{\hat{\sigma}^2_{p_{cy}}} \\ \frac{\sigma_{p_{cy}}}{\hat{\sigma}^2_{p_{cy}}} - \frac{1}{\sigma_{p_{cy}}} \end{bmatrix} + \frac{1}{M}\sum_{i=1}^{N}\sum_{k=1}^{M} \begin{bmatrix} \frac{({p_{cx}}^k - \mu_{p_{cx}})}{\sigma^2_{p_{cx}}} \\ \frac{\left(({p_{cx}}^k - \mu_{p_{cx}})^2 - \sigma^2_{p_{cx}}\right)}{\sigma^3_{p_{cx}}} \\ \frac{\left({p_{cy}}^k - \mu_{p_{cy}}\right)}{\sigma^2_{p_{cy}}} \\ \frac{\left(\left({p_{cy}}^k - \mu_{p_{cy}}\right)^2 - \sigma^2_{p_{cy}}\right)}{\sigma^3_{p_{cy}}} \end{bmatrix} \left(g_i(\mathbf{a}^k, p_c^k) + h_i(\mathbf{m}^k, \mathbf{s}^k, p_c^k)\right) \tag{B.29}$$

We define the eyeball radiis Jacobian $\mathbf{J_r}$ as follows:

$$\mathbf{J_r} := \begin{bmatrix} \frac{\partial \mathscr{L}}{\partial \mu_{r_e}} \\ \frac{\partial \mathscr{L}}{\partial \sigma_{r_e}} \\ \frac{\partial \mathscr{L}}{\partial \mu_{r_c}} \\ \frac{\partial \mathscr{L}}{\partial \sigma_{r_c}} \end{bmatrix} \approx - \begin{bmatrix} \frac{(\mu_{r_e} - \hat{\mu}_{r_e})}{\hat{\sigma}^2_{r_e}} \\ \frac{\sigma_{r_e}}{\hat{\sigma}^2_{r_e}} - \frac{1}{\sigma_{r_e}} \\ \frac{(\mu_{r_c} - \hat{\mu}_{r_c})}{\hat{\sigma}^2_{r_c}} \\ \frac{\sigma_{r_c}}{\hat{\sigma}^2_{r_c}} - \frac{1}{\sigma_{r_c}} \end{bmatrix} + \frac{1}{M} \sum_{i=1}^{N} \sum_{k=1}^{M} \begin{bmatrix} \frac{\left(r_e{}^k - \mu_{r_e}\right)}{\sigma^2_{r_e}} \\ \frac{\left(\left(r_e{}^k - \mu_{r_e}\right)^2 - \sigma^2_{r_e}\right)}{\sigma^3_{r_e}} \\ \frac{\left(r_c{}^k - \mu_{r_c}\right)}{\sigma^2_{r_c}} \\ \frac{\left(\left(r_c{}^k - \mu_{r_c}\right)^2 - \sigma^2_{r_c}\right)}{\sigma^3_{r_c}} \end{bmatrix} h_i(\mathbf{m}^k, \mathbf{s}^k, p_c^k) \tag{B.30}$$

Now, provided the result in Eq. B.22, we define the per sample orientation Jacobian $\mathbf{J}^i_{\mathbf{o}}$ as:

$$\mathbf{J}^i_{\mathbf{o}} := \begin{bmatrix} \frac{\partial \mathscr{L}}{\partial \mu_{\phi^i}} \\ \frac{\partial \mathscr{L}}{\partial \sigma_{\phi^i}} \\ \frac{\partial \mathscr{L}}{\partial \mu_{\theta^i}} \\ \frac{\partial \mathscr{L}}{\partial \sigma_{\theta^i}} \end{bmatrix} \approx - \begin{bmatrix} \frac{\left(\mu^i_\phi - \hat{\mu}^i_\phi\right)}{\hat{\sigma}^2_{\mathbf{o}}} \\ \frac{\sigma^i_\phi}{\hat{\sigma}^2_{\mathbf{o}}} - \frac{1}{\sigma^i_\phi} \\ \frac{(\mu^i_\theta - \hat{\mu}^i_\theta)}{\hat{\sigma}^2_{\mathbf{o}}} - \frac{a_u(\mu^i_{u_e} - (a_u \mu^i_\theta + b_u))}{\hat{\sigma}^2_{u_e}} \\ \frac{\sigma^i_\theta}{\hat{\sigma}^2_{\mathbf{o}}} - \frac{1}{\sigma^i_\theta} + \frac{a_u^2 \sigma^i_\theta}{\hat{\sigma}^2_{u_e}} \end{bmatrix} + \frac{1}{M} \sum_{k=1}^{M} \begin{bmatrix} \frac{\left(\phi^k - \mu^i_\phi\right)}{\sigma^{i\,2}_\phi} \\ \frac{\left(\left(\phi^k - \mu^i_\phi\right)^2 - \sigma^{i\,2}_\phi\right)}{\sigma^{i\,3}_\phi} \\ \frac{\left(\theta^k - \mu^i_\theta\right)}{\sigma^{i\,2}_\theta} \\ \frac{\left(\left(\theta^k - \mu^i_\theta\right)^2 - \sigma^{i\,2}_\theta\right)}{\sigma^{i\,3}_\theta}, \end{bmatrix} h_i(\mathbf{m}^k, \mathbf{s}^k, p_c^k), \tag{B.31}$$

where to evaluate the Monte Carlo expectations involved in the previous definitions we have drawn M samples $\{(\mathbf{m}^k, \mathbf{s}^k, \mathbf{a}^k, p_c^k)\}_{k=1}^M$ from the current distribution $q(\mathbf{m}^i)q(\mathbf{s})q(\mathbf{a})q(p_c)$, i.e., M samples per training sample i.

We finally define the group Jacobian as $\mathbf{J_g} = \left[\mathbf{J}^\top_{p_{cxy}}, \mathbf{J}^\top_{\mathbf{r}}, \mathbf{J}^{1\,\top}_{\mathbf{o}}, \ldots, \mathbf{J}^{N\,\top}_{\mathbf{o}}\right]^\top$ which is used to optimize jointly the eyeball geometry and the eye orientation per sample.

B.5.4 Axial and eyeball depth parameters

In this section we define the derivatives for the joint optimization of the axial parameters $\mathbf{a} := (\mathbf{N}, \phi_\kappa, \theta_\kappa)$ and the eyeball depth p_{cz}. We will refer this group of variables as $\mathbf{A} := (\mathbf{N}, \phi_\kappa, \theta_\kappa, p_{cz})$.

We thus identify the additives terms of $\ln p(\mathbf{X}, \mathbf{Z})$ including $\mathbf{A}$, as in Eq. B.9, we obtain the variational lower bound in function of only the terms related to $\mathbf{A}$:

$$\begin{aligned} \mathscr{L}(q) \approx \mathbb{E}_{\phi_\kappa}\left[\ln \mathcal{N}(\hat{\mu}_{\phi_\kappa}, \hat{\sigma}_{\phi_\kappa})\right] + \mathbb{E}_{\theta_\kappa}\left[\ln \mathcal{N}(\hat{\mu}_{\theta_\kappa}, \hat{\sigma}_{\theta_\kappa})\right] + \mathbb{E}_{\mathbf{N}}\left[\ln \mathcal{N}(\hat{\mu}_{\mathbf{N}}, \hat{\sigma}_{\mathbf{N}})\right] + \mathbb{E}_{p_{cz}}\left[\ln \mathcal{N}(\hat{\mu}_{p_{cz}}, \hat{\sigma}_{p_{cz}})\right] + \\ + \ln \sigma_{\phi_\kappa} + \ln \sigma_{\theta_\kappa} + \ln \sigma_{\mathbf{N}} + \ln \sigma_{p_{cz}} + \frac{1}{M} \sum_{i=1}^{N} \sum_{k=1}^{M} g_i(\mathbf{a}^k, p_c^k) + K_{\mathbf{A}} \end{aligned} \tag{B.32}$$

Where $K_{\mathbf{A}}$ is a constant as it contains the terms not related to $\mathbf{A}$. We can refer to Eq. B.19 to obtain the expectations for the terms related to the priors. In the case of the Jacobian, we develop Eq. B.11 and Eq. B.12, and taking into account the results shown in Eq. B.20 and Eq. B.21, we obtain $\mathbf{J_a}$ as follows:

$$\mathbf{J_A} := \begin{bmatrix} \frac{\partial \mathscr{L}}{\partial \mu_{\mathbf{N}}} \\ \frac{\partial \mathscr{L}}{\partial \sigma_{\mathbf{N}}} \\ \frac{\partial \mathscr{L}}{\partial \mu_{\phi_\kappa}} \\ \frac{\partial \mathscr{L}}{\partial \sigma_{\phi_\kappa}} \\ \frac{\partial \mathscr{L}}{\partial \mu_{\theta_\kappa}} \\ \frac{\partial \mathscr{L}}{\partial \sigma_{\theta_\kappa}} \\ \frac{\partial \mathscr{L}}{\partial \mu_{pcz}} \\ \frac{\partial \mathscr{L}}{\partial \sigma_{pcz}} \end{bmatrix} \approx - \begin{bmatrix} \frac{(\mu_{\mathbf{N}} - \hat{\mu}_{\mathbf{N}})}{\hat{\sigma}^2_{\mathbf{N}}} \\ \frac{\sigma_{\mathbf{N}}}{\hat{\sigma}^2_{\mathbf{N}}} - \frac{1}{\sigma_{\mathbf{N}}} \\ \frac{(\mu_{\phi_\kappa} - \hat{\mu}_{\phi_\kappa})}{\hat{\sigma}^2_{\phi_\kappa}} \\ \frac{\sigma_{\phi_\kappa}}{\hat{\sigma}^2_{\phi_\kappa}} - \frac{1}{\sigma_{\phi_\kappa}} \\ \frac{(\mu_{\theta_\kappa} - \hat{\mu}_{\theta_\kappa})}{\hat{\sigma}^2_{\theta_\kappa}} \\ \frac{\sigma_{\theta_\kappa}}{\hat{\sigma}^2_{\theta_\kappa}} - \frac{1}{\sigma_{\theta_\kappa}} \\ \frac{(\mu_{pcz} - \hat{\mu}_{pcz})}{\hat{\sigma}^2_{pcz}} \\ \frac{\sigma_{pcz}}{\hat{\sigma}^2_{pcz}} - \frac{1}{\sigma_{pcz}} \end{bmatrix} + \frac{1}{M} \sum_{i=1}^{N} \sum_{k=1}^{M} \begin{bmatrix} \frac{(\mathbf{N}^k - \mu_{\mathbf{N}})}{\sigma^2_{\mathbf{N}}} \\ \frac{\left((\mathbf{N}^k - \mu_{\mathbf{N}})^2 - \sigma^2_{\mathbf{N}}\right)}{\sigma^3_{\mathbf{N}}} \\ \frac{({\phi_\kappa}^k - \mu_{\phi_\kappa})}{\sigma^2_{\phi_\kappa}} \\ \frac{\left(({\phi_\kappa}^k - \mu_{\phi_\kappa})^2 - \sigma^2_{\phi_\kappa}\right)}{\sigma^3_{\phi_\kappa}} \\ \frac{({\theta_\kappa}^k - \mu_{\theta_\kappa})}{\sigma^2_{\theta_\kappa}} \\ \frac{\left(({\theta_\kappa}^k - \mu_{\theta_\kappa})^2 - \sigma^2_{\theta_\kappa}\right)}{\sigma^3_{\theta_\kappa}} \\ \frac{({p_{cz}}^k - \mu_{pcz})}{\sigma^2_{pcz}} \\ \frac{\left(({p_{cz}}^k - \mu_{pcz})^2 - \sigma^2_{pcz}\right)}{\sigma^3_{pcz}} \end{bmatrix} g_i(\mathbf{a}^k, p_c^k) \tag{B.33}$$

Where in order to evaluate Eq. B.32 and Eq. B.33 we have drawn M samples $\{(\mathbf{a}^k, p_c^k)\}_{k=1}^M$ from the current distribution $q(\mathbf{a})q(p_c)$ to use a Monte Carlo expectation approximation.

B.6 Efficient sampling: semi-integral likelihoods

Notice that Eq. B.18 involves an integration over the image according to the segmentation parametrized by $(\mathbf{m}, \mathbf{s}, p_c)$. In order to compute such integration faster we can precompute semi-integral images as follows:

$$\mathscr{I}(u, v) = \sum_{j=1}^{v} (1 - \omega^{u,j}) \ln p(c^{u,j} | \Lambda_l) \tag{B.34}$$

Given that the segmentation is parametric, we can query per image column coordinate the boundaries of each region (cornea, sclera and skin) as row pixel coordinates. We then evaluate the integral of the image column by computing differences of $\mathscr{I}$ at the boundaries.

This is highly valuable as likelihood image integrations from Eq. B.18 are used throughout the inference process. In particular, it is where the main computational cost resides during test time.

This process, combined with a GPU implementation of the Monte Carlo expectation, leads to a fast implementation of the movement parameters inference at test phase.

Bibliography

B Abboud and F Davoine. Bilinear factorization for facial expression analysis and synthesis. *IEE Proceedings - Vision, Image & Signal Processing*, 152(3):327–333, June 2005.

B Amberg, S Romdhani, and T Vetter. Optimal Step Nonrigid ICP Algorithms for Surface Registration. In *IEEE Computer Vision and Pattern Recognition*, pages 1–8, 2007.

Brian Amberg, Reinhard Knothe, and Thomas Vetter. Expression invariant 3D face recognition with a Morphable Model. In *Int. Conf. on Automatic Face and Gesture Recognition*, pages 1–6. IEEE, September 2008.

Sean Andrist, Xiang Zhi Tan, Michael Gleicher, and Bilge Mutlu. Conversational gaze aversion for humanlike robots. *International Conference on Human-Robot Interaction*, pages 25–32, 2014.

S Ba and J-M. Odobez. A probabilistic framework for joint head tracking and pose estimation. In *17th Int. Conf. Pattern Recognition (ICPR)*, volume 4, Cambridge, UK, August 2004.

S Ba and J M Odobez. A Study on Visual Focus of Attention Recognition from Head Pose in a Meeting Room. In *Proc. Workshop on Machine Learning for Multimodal Interaction (MLMI)*, Washington DC, May 2006.

Simon Baker and Iain Matthews. Lucas-Kanade 20 Years On: A Unifying Framework. *International Journal of Computer Vision*, 56(3):221–255, 2004.

Tadas Baltrusaitis, Peter Robinson, and L.P. Morency. 3D Constrained Local Model for Rigid and Non-Rigid Facial Tracking. In *Computer Vision and Pattern Recognition (CVPR)*, Providence, RI, June 2012.

Shumeet Baluja and Dean Pomerleau. Non-Intrusive Gaze Tracking Using Artificial Neural Networks. Technical report, CMU, 1994.

Jonathan Barron and Jitendra Malik. Shape, Illumination, and Reflectance from Shading. Technical Report UCB/EECS-2013-117, University of California, Berkeley, May 2013.

S Basu, I Essa, and A Pentland. Motion Regularization for Model-Based Head Tracking. In *International Conference on Pattern Recognition*, ICPR '96, pages 611—-, Washington, DC, USA, 1996. IEEE Computer Society.

Chiraz BenAbdelkader. Robust Head Pose Estimation Using Supervised Manifold Learning. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *European Conference on Computer Vision*, volume 6316 of *Lecture Notes in Computer Science*, pages 518–531. Springer Berlin Heidelberg, 2010.

Pascal Bérard, Derek Bradley, Maurizio Nitti, Thabo Beeler, and Markus Gross. High-quality capture of eyes. *ACM Transactions on Graphics*, 33(6):1–12, November 2014. ISSN 07300301. doi: 10.1145/2661229.2661285. URL http://dl.acm.org/citation.cfm?id=2661229.2661285.

P J Besl and Neil D McKay. A method for registration of 3-D shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(2):239–256, February 1992.

D J Beymer. Face recognition under varying pose. In *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, pages 756–761, June 1994.

Timothy W Bickmore and Rosalind W Picard. Establishing and maintaining long-term human-computer relationships. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 12 (2):293–327, 2005.

Christopher M Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, October 2007. ISBN 0387310738.

V. Blanz and T. Vetter. Face recognition based on fitting a 3D morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1063–1074, September 2003.

Ali Borji and Laurent Itti. State-of-the-art in visual attention modeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):185–207, January 2013.

Sofien Bouaziz, Yangang Wang, and Mark Pauly. Online Modeling for Realtime Facial Animation. *ACM Trans. Graph.*, 32(4):40:1—-40:10, July 2013.

C Bregler and J Malik. Tracking people with twists and exponential maps. In *Computer Vision and Pattern Recognition*, pages 8–15, 1998.

Zoya Bylinskii, Tilke Judd, Ali Borji, Laurent Itti, Frédo Durand, Aude Oliva, and Antonio Torralba. MIT Saliency Benchmark. http://saliency.mit.edu/.

Xudong Cao, Yichen Wei, Fang Wen, and Jian Sun. Face Alignment by Explicit Shape Regression. *International Journal of Computer Vision*, 2013.

M La Cascia, S Sclaroff, and V Athitsos. Fast, Reliable Head Tracking under Varying Illumination: An Approach Based on Registration of Texture-Mapped 3D Models. In *IEEE Trans. Pattern Anal. Machine Intell.*, volume 22, 2000.

Justine Cassell. Nudge nudge wink wink: Elements of face-to-face conversation for embodied conversational agents. *Embodied conversational agents*, pages 1–27, 2000.

Justine Cassell, Hannes Högni Vilhjálmsson, and Timothy Bickmore. BEAT: The Behavior Expression Animation Toolkit. In *Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 477–486, New York, NY, USA, 2001.

Y Chen and G Medioni. Object modeling by registration of multiple range images. In *International Conference on Robotics and Automation*, pages 2724–2729 vol.3, April 1991.

Doo Hyun Choi, Ick Hoon Jang, Mi Hye Kim, and Nam Chul Kim. Color Image Enhancement Based on Single-Scale Retinex With a JND-Based Nonlinear Filter. In *ISCAS*, pages 3948–3951. IEEE, 2007.

T F Cootes, C J Taylor, D H Cooper, and J Graham. Active Shape Models - their Training and Application. *Computer Vision and Image Understanding*, 61(1):38–59, January 1995.

T F Cootes, G J Edwards, and C J Taylor. Active Appearance Models. In *Proceedings of the European Conference on Computer Vision*, volume 2, pages 183–191, November 1998.

T F Cootes, K Walker, and C J Taylor. View-Based Active Appearance Models. In *Automatic Face and Gesture Recognition*, pages 227–232. IEEE Society, IEEE, 2000.

Darren Cosker, Eva Krumhuber, and Adrian Hilton. A FACS valid 3D dynamic action unit database with applications to 3D dynamic morphable facial modeling. In *IEEE International Conference on Computer Vision (ICCV)*, Barcelona, Spain, November 2011.

Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. In *Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 886–893, June 2005.

M Dantone, J Gall, G Fanelli, and L Van Gool. Real-time Facial Feature Detection using Conditional Regression Forests. In *Computer Vision and Pattern Recognition*, 2012.

D.Cristinacce and T.F.Cootes. Automatic Feature Localisation with Constrained Local Models. *Pattern Recognition*, 41(10):3054–3067, 2007.

F Dornaika and F Davoine. Simultaneous facial action tracking and expression recognition in the presence of head motion. *International Journal of Computer Vision*, 76(3):257–281, March 2008.

Andrew T Duchowski. *Eye Tracking Methodology: Theory and Practice*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007. ISBN 1846286085.

Starkey Duncan. Some signals and rules for taking speaking turns in conversations. *Journal of Personality and Social Psychology*, 23:283–292, 1972.

Bernhard Egger, Sandro Schonborn, Andreas Forster, and Thomas Vetter. Pose Normalization for Eye Gaze Estimation and Facial Attribute Description from Still Images. In *German Conference on Pattern Recognition*, 2014.

P Ekman and W Friesen. *Facial Action Coding System: A Technique for the Measurement of Facial Movement.* Consulting Psychologists Press, Palo Alto, 1978.

EyeGaze. http://www.eyegaze.com, 2005.

G. Fanelli and J. Gall. Real Time Head Pose Estimation with Random Regression Forests. *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, June 2011.

G Fanelli, T Weise, J Gall, and L Van Gool. Real Time Head Pose Estimation from Consumer Depth Cameras. In *Symposium of the German Association for Pattern Recognition (DAGM)*, September 2011.

L. Fletcher and A. Zelinsky. Driver Inattention Detection based on Eye Gaze-Road Event Correlation. *The International Journal of Robotics Research*, 28(6):774–801, 2009.

Simone Frintrop, Erich Rome, and Henrik I. Christensen. Computational visual attention systems and their cognitive foundations. *ACM Transactions on Applied Perception*, 7(1): 1–39, 2010.

Kenneth Alberto Funes Mora and Jean-Marc Odobez. Gaze Estimation From Multimodal Kinect Data. In *Computer Vision and Pattern Recognition, Workshop on Gesture Recognition*, pages 25–30, June 2012.

Kenneth Alberto Funes Mora and Jean-Marc Odobez. Person Independent 3D Gaze Estimation From Remote RGB-D Cameras. In *IEEE Int. Conf. on Image Processing*, September 2013.

Kenneth Alberto Funes Mora and Jean-Marc Odobez. Geometric generative gaze estimation (G3E) for remote RGB-D cameras. In *Computer Vision and Pattern Recognition*, Ohio, June 2014a.

Kenneth Alberto Funes Mora and Jean-Marc Odobez. 3D Gaze Tracking and Automatic Gaze Coding from RGB-D Cameras. In *Computer Vision and Pattern Recognition, Vision Meets Cognition Workshop*, June 2014b.

Kenneth Alberto Funes Mora and Jean-Marc Odobez. Pattent Application PCT/EP2014/062604: A Gaze Estimation Method and apparatus, 2014c.

Kenneth Alberto Funes Mora and Jean Marc Odobez. Gaze Estimation in the 3D Space Using RGB-D sensors. Towards Head-Pose And User Invariance. *Journal Paper Under Review*, 2015.

Kenneth Alberto Funes Mora, Laurent Son Nguyen, Daniel Gatica-Perez, and Jean-Marc Odobez. A Semi-Automated System for Accurate Gaze Coding in Natural Dyadic Interactions. In *Int. Conf. on Multimodal Interaction*, December 2013.

Kenneth Alberto Funes Mora, Florent Monay, and Jean-Marc Odobez. EYEDIAP: A Database for the Development and Evaluation of Gaze Estimation Algorithms from RGB and RGB-D Cameras. In *Symposium on Eye tracking Research & Applications*, 2014a.

Kenneth Alberto Funes Mora, Florent Monay, and Jean-Marc Odobez. EYEDIAP Database: Data Description and Gaze Tracking Evaluation Benchmarks. Technical Report Idiap-RR-08-2014, Idiap, 2014b.

D. Gatica-Perez, I. McCowan, and S. Bengio. Detecting Group Interest-Level in Meetings. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 489–492, March 2005.

Andrew Gee and Roberto Cipolla. Determining the gaze of faces in images. *Image and Vision Computing*, 12(10):639–647, December 1994.

Andrew Gee and Roberto Cipolla. Fast visual tracking by temporal consensus. *Image and Vision Computing*, 14(2):105–114, March 1996. ISSN 02628856. doi: 10.1016/0262-8856(95)01044-0.

Burak S Göktürk, J Y Bouguet, and R Grzeszczuk. A Data-Driven Model for Monocular Face Tracking. In *Proc. 8th International Conference on Computer Vision, Vancouver, Canada*, volume 2, pages 701–708, 2001.

Sebastian Gorga and Kazuhiro Otsuka. Conversation scene analysis based on dynamic Bayesian network and image-based gaze detection. In *International Conference on Multimodal Interfaces*, pages 8–12, 2010.

R Gross, I Matthews, and S Baker. Active Appearance Models with Occlusion. *Image & Vision Computing Journal*, 24(6):593–604, 2006.

Ralph Gross, Iain Matthews, and Simon Baker. Generic vs. person specific active appearance models. *Image and Vision Computing*, 23(1):1080–1093, November 2005.

Elias D Guestrin and Moshe Eizenman. Listing's and Donders' laws and the estimation of the point-of-gaze. In *Symp. on Eye Tracking Research & Applications*, Austin, TX, 2010.

Elias Daniel Guestrin and Moshe Eizenman. General theory of remote gaze estimation using the pupil center and corneal reflections. *Transactions on bio-medical engineering*, 53(6):1124–33, June 2006.

G D Hager and P N Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(10):1025–1039, October 1998.

Dan Witzner Hansen and Qiang Ji. In the eye of the beholder: a survey of models for eyes and gaze. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(3):478–500, March 2010.

Dan Witzner Hansen and Arthur E.C. C Pece. Eye tracking in the wild. *Computer Vision and Image Understanding*, 98(1):155–181, April 2005.

Dan Witzner Hansen, Javier San Agustin, and Arantxa Villanueva. Homography Normalization for Robust Gaze Estimation in Uncalibrated Setups. In *Symposium on Eye-Tracking Research & Applications*, pages 13–20, New York, NY, USA, 2010. ACM.

D.W. Hansen, J.P. Hansen, M. Nielsen, A.S. Johansen, and M.B. Stegmann. Eye typing using Markov and active appearance models. *IEEE Workshop on Applications of Computer Vision*, pages 132–136, 2002.

Daniel Herrera C., Juho Kannala, and Janne Heikkilä. Joint Depth and Color Camera Calibration with Distortion Correction. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 34 (10):2058–2064, 2012.

T Horprasert, Y Yacoob, and L Davis. Computing 3D Head Orientation from a Monocular Image Sequence. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 1996.

T Horprasert, Y Yacoob, and L S Davis. An Anthropometric Shape Model For Estimating Head Orientation. In *In Proceedings of the Third International Workshop on Visual Form*, 1997.

J Huang, Xuhui Shao, and H Wechsler. Face pose discrimination using support vector machines (SVM). In *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, volume 1, pages 154–156 vol.1, August 1998. doi: 10.1109/ICPR.1998.711102.

Hayley Hung, D.B. Jayagopi, Sileye Ba, J.M. Odobez, and D. Gatica-Perez. Investigating automatic dominance estimation in groups from visual attention and speaking activity. In *Int. Conf. on Multimodal Interfaces*, 2008.

Kazushi Hyoki, Masahiro Shigeta, Norifumi Tsuno, Yu Kawamuro, and Toshihiko Kinoshita. Quantitative electro-oculography and electroencephalography as indices of alertness. *Electroencephalography and Clinical Neurophysiology*, 106(3):213–219, March 1998.

Takahiro Ishikawa, Simon Baker, Iain Matthews, and Takeo Kanade. Passive Driver Gaze Tracking with Active Appearance Models. In *Proc. World Congress on Intelligent Transportation Systems*, pages 1–12, October 2004.

Poika Isokoski, Markus Joos, Oleg Spakov, and Benoét Martin. Gaze controlled games. *Universal Access in the Information Society*, 8(4):323–337, 2009. ISSN 16155289. doi: 10.1007/s10209-009-0146-3.

Jun-Su Jang and Takeo Kanade. Robust 3D Head Tracking by Online Feature Registration. In *The IEEE International Conference on Automatic Face and Gesture Recognition*, September 2008.

Dinesh Babu Jayagopi, Samira Sheikhi, David Klotz, Johannes Wienke, Jean-Marc Odobez, Sebastian Wrede, Vasil Khalidov, Laurent Son Nguyen, Britta Wrede, and Daniel Gatica-Perez. The vernissage corpus: a conversational human-robot-interaction dataset. In *Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction*, 2013.

Dineshbabu Jayagopi, Dairazalia Sanchez-Cortes, Kazuhiro Otsuka, Junji Yamato, and Daniel Gatica-Perez. Linking speaking and looking behavior patterns with group composition, perception, and performance. In *ICMI*, page 433, New York, 2012.

T S Jebara and A Pentland. Parametrized Structure from Motion for 3D Adaptive Feedback Tracking of Faces. In *Computer Vision and Pattern Recognition (CVPR)*, CVPR '97, pages 144—-, Washington, DC, USA, 1997. IEEE Computer Society.

Li Jianfeng and Li Shigang. Eye-Model-Based Gaze Estimation by RGB-D Camera. In *Computer Vision and Pattern Recognition Workshops*, pages 606–610, June 2014.

Vahid Kazemi and Josephine Sullivan. One Millisecond Face Alignment with an Ensemble of Regression Trees. In *Computer Vision and Pattern Recognition (CVPR)*, 2014.

Adam Kendon. Some functions of gaze-direction in social interaction. *Acta Psychologica*, 26: 22–63, January 1967.

M L Knapp and J A Hall. *Nonverbal communication in human interaction*. Wadsworth, Cengage Learning, 7 edition, 2009. ISBN 9780495568698.

Norbert Krüger, Michael Pötzsch, and Christoph von der Malsburg. Determination of face position and pose with a learned representation based on labelled graphs. *Image Vision Comput.*, 15(8):665–673, 1997.

Stephen R H Langton, Helen Honeyman, and Emma Tessler. The influence of head contour and nose angle on the perception of eye-gaze direction. *Perception Psychophysics*, 66(5): 752–771, 2004.

A Lanitis, C J Taylor, T Ecootes, and T Ahmed. Automatic interpretation of human faces and hand gestures using flexible models. In *In International Workshop on Automatic Face- and Gesture-Recognition*, pages 98–103, 1995.

Andreas Lanitis, Chris J. Taylor, and Timothy F. Cootes. Automatic interpretation and coding of face images using flexible models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):743–756, 1997.

Stéphanie Lefèvre and Jean-Marc Odobez. Structure and appearance features for robust 3D facial actions tracking. In *IEEE International Conference on Multimedia and Expo*, ICME'09, pages 298–301, Piscataway, NJ, USA, June 2009. IEEE Press.

V Lepetit, J Pilet, and P Fua. Point matching as a classification problem for fast and robust object pose estimation. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–244–II–250 Vol.2, June 2004. doi: 10.1109/CVPR.2004.1315170.

Bruno Lepri, R Subramanian, and K Kalimeri. Employing social gaze and speaking activity for automatic determination of the extraversion trait. In *ICMI*, 2010.

Dongheng Li, D Winfield, and D J Parkhurst. Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. In *Computer Vision and Pattern Recognition Workshops*, volume 3, page 79. IEEE, June 2005.

Hao Li, Thibaut Weise, and Mark Pauly. Example-based facial rigging. *ACM Trans. Graph.*, 29 (4):32:1—-32:6, July 2010.

W K Liao and G Medioni. 3D Face Tracking and Expression Inference from a 2D Sequence using Manifold Learning. In *Proc. Comp. Vision and Pattern Recognition conference*, 2008.

David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.

Feng Lu, Yusuke Sugano, Okabe Takahiro, Yoichi Sato, and Takahiro Okabe. Inferring Human Gaze from Appearance via Adaptive Linear Regression. In *International Conference on Computer Vision (ICCV)*, November 2011a.

Feng Lu, Okabe Takahiro, Yusuke Sugano, and Yoichi Sato. A Head Pose-free Approach for Appearance-based Gaze Estimation. In *British Machine Vision Conference (BMVC)*, 2011b.

Feng Lu, Y Sugano, T Okabe, and Y Sato. Head pose-free appearance-based gaze sensing via eye image synthesis. In *IEEE International Conference on Pattern Recognition*, November 2012.

Paivi Majaranta, Hirotaka Aoki, Mick Donegan, Dan Witzner Hansen, and John Paulin Hansen. *Gaze Interaction and Applications of Eye Tracking: Advances in Assistive Technologies*. Information Science Reference - Imprint of: IGI Publishing, Hershey, PA, 1st edition, 2011. ISBN 161350098X, 9781613500989.

Marius Malciu and Françoise Prêteux. A Robust Model-Based Approach for 3D Head Tracking in Video Sequences. In *International Conference on Automatic Face and Gesture Recognition*, FG '00, pages 169—-, Washington, DC, USA, 2000.

Francis Martinez, Andrea Carbone, and Edwige Pissaloux. Gaze estimation using local features and non-linear regression. In *Int. Conf. on Image Processing*, pages 1961–1964, September 2012.

Iain Matthews and Simon Baker. Active Appearance Models Revisited. *International Journal of Computer Vision*, 60(1):135–164, November 2004.

John Merchant, Richard Morrissette, and James L Porterfield. Remote Measurement of Eye Direction Allowing Subject Motion Over One Cubic Foot of Space. *Biomedical Engineering, IEEE Transactions on*, BME-21(4):309–317, July 1974.

L P Morency, A Rahimi, and T Darrell. Adaptive View-based Appearance Model. In *IEEE Conf. Comp. Vision and Pattern Recognition*, 2003a.

L.P. Morency and T. Darrell. Stereo tracking using ICP and normal flow constraint. In *Int. Conf. on Pattern Recognition,*, volume 4, pages 367–372. IEEE, 2002.

L.P. Morency, A Rahimi, N Checka, and T Darrell. Fast stereo-based head tracking for interactive environments. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pages 390–395, May 2002.

L.P. Morency, P Sundberg, and T Darrell. Pose estimation using 3D view-based eigenspaces. In *Analysis and Modeling of Faces and Gestures, 2003. AMFG 2003. IEEE International Workshop on*, pages 45–52, October 2003b.

L.P. Morency, J Whitehill, and J Movellan. Generalized Adaptive View-based Appearance Model: Integrated Framework for Monocular Head Pose Estimation. In *IEEE Int. Conf. on Automatic Face and Gesture Recognition (FG)*, 2008.

L.P. Morency, Jacob Whitehill, and Javier Movellan. Monocular Head Pose Estimation Using Generalized Adaptive View-based Appearance Model. *Image Vision Comput.*, 28(5):754–761, May 2010.

C H Morimoto and M R M Mimica. Eye gaze tracking techniques for interactive applications. *Computer Vision and Image Understanding*, 98:4–24, 2005.

C.H Morimoto, D Koons, A Amir, and M Flickner. Pupil detection and tracking using multiple light sources. *Image and Vision Computing*, 18(4):331–335, March 2000.

T. Moriyama and J.F. Cohn. Meticulously detailed eye model and its application to analysis of facial image. *Int. Conf. on Systems, Man and Cybernetics*, 1:629–634, 2004.

E Murphy-Chutorian and M Trivedi. Head Pose Estimation in Computer Vision: A Survey. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2008a.

E Murphy-Chutorian and M M Trivedi. HyHOPE: Hybrid Head Orientation and Position Estimation for vision-based driver head tracking. In *Intelligent Vehicles Symposium, 2008 IEEE*, pages 512–517, June 2008b.

E Murphy-Chutorian, A Doshi, and M M Trivedi. Head Pose Estimation for Driver Assistance Systems: A Robust Algorithm and Experimental Evaluation. In *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pages 709–714, September 2007.

B Mutlu, J Forlizzi, and J Hodgins. A Storytelling Robot: Modeling and Evaluation of Human-like Gaze Behavior. In *International Conference on Humanoid Robots*, pages 518–523, December 2006.

Basilio Noris, Karim Benmachiche, and Aude Billard. Calibration-Free Eye Gaze Direction Detection with Gaussian Processes. In *VISAPP (2)*, pages 611–616, 2008.

Basilio Noris, J.B. Keller, and Aude Billard. A wearable gaze tracking system for children in unconstrained environments. *Computer Vision and Image Understanding*, pages 1–27, 2010.

Catharine Oertel and Giampiero Salvi. A gaze-based method for relating group involvement to individual engagement in multimodal multiparty dialogue. In *International Conference on Multimodal Interaction (ICMI)*, pages 99–106, New York, New York, USA, December 2013. ACM Press. doi: 10.1145/2522848.2522865.

Catharine Oertel, Kenneth Alberto Funes Mora, Samira Sheikhi, Jean-Marc Odobez, and Joakim Gustafson. Who Will Get the Grant? A Multimodal Corpus for the Analysis of Conversational Behaviours in Group. In *International Conference on Multimodal Interaction, Understanding and Modeling Multiparty, Multimodal Interactions Workshop*, November 2014.

Kenji Oka, Yoichi Sato, Yasuto Nakanishi, and Hideki Koike. Head Pose Estimation System Based on Particle Filtering with Adaptive Diffusion Control. In *MVA*, pages 586–589, 2005.

Manfred Opper and Cédric Archambeau. The variational gaussian approximation revisited. *Neural Comput.*, March 2009.

Margarita Osadchy, Yann Le Cun, and Matthew L Miller. Synergistic Face Detection and Pose Estimation with Energy-Based Models. *J. Mach. Learn. Res.*, 8:1197–1215, May 2007. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1248659.1248700.

R Pappu and P A Beardsley. A qualitative approach to classifying gaze direction. In *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pages 160–165, April 1998. doi: 10.1109/AFGR.1998.670942.

Soon-Yong Park and Murali Subbarao. An accurate and fast point-to-plane registration technique. *Pattern Recogn. Lett.*, 24(16):2967–2976, December 2003. ISSN 0167-8655. doi: 10.1016/S0167-8655(03)00157-0. URL http://dx.doi.org/10.1016/S0167-8655(03)00157-0.

P Paysan, R Knothe, B Amberg, S Romdhani, and T Vetter. A 3D Face Model for Pose and Illumination Invariant Face Recognition. In *Proceedings of Advanced Video and Signal based Surveillance*. IEEE, 2009.

F Pedregosa, G Varoquaux, A Gramfort, V Michel, B Thirion, O Grisel, M Blondel, P Prettenhofer, R Weiss, V Dubourg, J Vanderplas, A Passos, D Cournapeau, M Brucher, M Perrot, and E Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Krishnan Ramnath, Simon Baker, Iain Matthews, and Deva Ramanan. Increasing the density of Active Appearance Models. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, June 2008. doi: 10.1109/CVPR.2008.4587517. URL http://dx.doi.org/10.1109/CVPR.2008.4587517.

B Raytchev, I Yoda, and K Sakaue. Head pose estimation by nonlinear manifold learning. In *International Conference on Pattern Recognition (ICPR)*, volume 4, pages 462–466 Vol.4, August 2004.

J M Rehg, G D Abowd, A Rozga, M Romero, M A Clements, S Sclaroff, I Essa, O Y Ousley, Yin Li, Chanho Kim, H Rao, J C Kim, L L Presti, Jianming Zhang, D Lantsman, J Bidwell, and Zhefan Ye. Decoding Children's Social Behavior. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3414–3421, June 2013.

Knothe Reinhard. *A Global-to-Local Model for the Representation of Human Faces*. PhD thesis, University of Basel, 2009.

H A Rowley, S Baluja, and T Kanade. Rotation invariant neural network-based face detection. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pages 38–44, June 1998.

David Rozado, Ahmad El Shoghri, and Raja Jurdak. Gaze dependant prefetching of web content to increase speed and comfort of web browsing. *International Journal of Human-Computer Studies*, 78:31–42, June 2015. doi: 10.1016/j.ijhcs.2015.02.006.

S. Rusinkiewicz and Marc Levoy. Efficient variants of the ICP algorithm. In *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pages 145–152. IEEE, 2001.

M Rydfalk. CANDIDE: A Parameterized Face. 1987.

Timo Schneider, Boris Schauerte, and Rainer Stiefelhagen. Manifold Alignment for Person Independent Appearance-based Gaze Estimation. In *International Conference on Pattern Recognition (ICPR)*. IEEE, August 2014.

Arno Schödl, Antonio Haro, and Irfan A Essa. Head Tracking Using a Textured Polygonal Model. In *In PUI98*, pages 43–48, 1998.

Elizabeth R Schotter, Raymond W Berry, Craig R M McKenzie, and Keith Rayner. Gaze bias: Selective encoding and liking effects. *Visual Cognition*, 18(8):1113–1132, 2010.

J. Sherrah, S. Gong, and E.J. Ong. Face distributions in similarity space under varying head pose. *Image and Vision Computing*, 19(12):807–819, October 2001.

Jamie Sherrah and Shaogang Gong. Fusion of perceptual cues for robust tracking of head pose and position. *Pattern Recognition*, 34(8):1565–1572, 2001.

Sheng-Wen Shih, Yu-Te Wu, and Jin Liu. A calibration-free gaze tracking technique. In *International Conference on Pattern Recognition*, volume 4, pages 201–204 vol.4, 2000. doi: 10.1109/ICPR.2000.902895.

Shinsuke Shimojo, Claudiu Simion, Eiko Shimojo, and Christian Scheier. Gaze bias both reflects and influences preference. *Nat Neurosci*, 6(12):1317–1322, December 2003. ISSN 1097-6256.

J Shotton, A Fitzgibbon, M Cook, T Sharp, M Finocchio, R Moore, A Kipman, and A Blake. Real-time Human Pose Recognition in Parts from Single Depth Images. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1297–1304, 2011.

Kohsia S.Huang and M M Trivedi. Robust real-time detection, tracking, and pose estimation of faces in video streams. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 965–968 Vol.3, August 2004.

SMI. http://www.smi.de, 2007.

Brian A Smith, Qi Yin, Steven K Feiner, and Shree K Nayar. Gaze Locking: Passive Eye Contact Detection for Human-object Interaction. In *Symposium on User Interface Software and Technology*, UIST '13, New York, NY, USA, 2013. ACM.

Alex J Smola and Bernhard Schölkopf. A Tutorial on Support Vector Regression. *Statistics and Computing*, 14(3):199–222, August 2004. ISSN 0960-3174.

Nikolai Smolyanskiy, Christian Huitema, Lin Liang, and Sean Eron Anderson. Real-time 3D face tracking based on active appearance model constrained by depth data. *Image and Vision Computing*, 32(11):860–869, November 2014.

S Srinivasan and K L Boyer. Head Pose Estimation using View Based Eigenspaces. In *International Conference on Pattern Recognition*, 2002.

Jacob Ström. Model-based Real-time Head Tracking. *EURASIP J. Appl. Signal Process.*, 2002(1): 1039–1052, January 2002.

Yusuke Sugano, Yasuyuki Matsushita, Yoichi Sato, and Hideki Koike. An incremental learning method for unconstrained gaze estimation. In *European Conference on Computer Vision (ECCV)*, pages 656–667. Springer, 2008.

Yusuke Sugano, Yasuyuki Matsushita, and Yoichi Sato. Learning-by-Synthesis for Appearance-based 3D Gaze Estimation. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014.

Jaewon Sung, Takeo Kanade, and Daijin Kim. Pose Robust Face Tracking by Combining Active Appearance Models and Cylinder Head Models. *International Journal on Computer Vision*, 80(2):260–274, November 2008.

Kar-Han Tan, David J Kriegman, and Narendra Ahuja. Appearance-based Eye Gaze Estimation. In *IEEE Workshop on Applications of Computer Vision*, pages 191—-, 2002.

Fabian Timm and Erhardt Barth. Accurate Eye Centre Localisation by Means of Gradients. In *Int. Conf. on Computer Vision Theory and Applications*, pages 125–130, 2011.

M A Turk and A P Pentland. Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition. IEEE Computer Society Conference on*, pages 586–591, June 1991.

D Tweed and T Vilis. Geometric relations of eye position and velocity vectors during saccades. *Vision Res*, 30(1):111–127, 1990.

L Vacchetti, V Lepetit, and P Fua. Fusing online and offline information for stable 3D tracking in real-time. In *Computer Vision and Pattern Recognition (CVPR)*, volume 2, June 2003.

L Vacchetti, V Lepetit, and P Fua. Stable Real-Time 3D Tracking using Online and Offline Information. *IEEE Trans. Pattern Anal. Machine Intell.*, 26(10):1385–1391, 2004.

R Valenti, N Sebe, and T Gevers. Combining Head Pose and Eye Location Information for Gaze Estimation. *IEEE Transactions on Image Processing*, 21(2):802–815, February 2012.

Roberto Valenti and Theo Gevers. Accurate eye center location through invariant isocentric patterns. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 34(9):1785–1798, September 2012. ISSN 1939-3539.

Thomas Vetter and Volker Blanz. Estimating Coloured 3D Face Models from Single Images: An Example Based Approach. In *In Proceedings, European Conference on Computer Vision*, volume II, pages 499–513. Springer, 1998.

M Viola, Michael J Jones, and Paul Viola. Fast Multi-view Face Detection. In *Proc. of Computer Vision and Pattern Recognition*, 2003.

Paul Viola and Michael Jones. Robust Real-time Object Detection. In *International Journal of Computer Vision*, 2001.

C Vogler, Z Li, A Kanaujia, S Goldenstein, and D Metaxas. The Best of Both Worlds: Combining 3D Deformable Models with Active Shape Models. In *Proceedings of the IEEE XXI International Conference on Computer Vision (ICCV)*, 2007.

Michael Voit, Kai Nickel, and Rainer Stiefelhagen. Neural Network-based Head Pose Estimation and Multi-view Fusion. In *International Evaluation Conference on Classification of Events, Activities and Relationships*, CLEAR'06, pages 291–298, Berlin, Heidelberg, 2007. Springer-Verlag.

Chao Wang and Xubo Song. Robust head pose estimation via supervised manifold learning. *Neural networks : the official journal of the International Neural Network Society*, 53:15–25, May 2014. ISSN 1879-2782. doi: 10.1016/j.neunet.2014.01.009.

Haibo Wang, F Davoine, V Lepetit, C Chaillou, and Chunhong Pan. 3D Head Tracking via Invariant Keypoint Learning. *Circuits and Systems for Video Technology, IEEE Transactions on*, 22(8):1113–1126, August 2012.

J.-G. Wang and E Sung. Study on eye gaze estimation. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 32(3):332–350, 2002.

Jian-Gang Wang and Eric Sung. Pose determination of human faces by using vanishing points. *Pattern Recognition*, 34(12):2427–2445, December 2001.

Jian-Gang Wang and Eric Sung. EM Enhancement of 3D Head Pose Estimated by Point at Infinity. *Image Vision Comput.*, 25(12):1864–1874, December 2007.

Thibaut Weise, Sofien Bouaziz, Hao Li, and Mark Pauly. Realtime performance-based facial animation. *ACM Transactions on Graphics (SIGGRAPH 2011)*, 30(4):1, July 2011. ISSN 07300301.

Amy M. Wetherby, Juliann Woods, Lori Allen, Julie Cleary, Holly Dickinson, and Catherine Lord. Early Indicators of Autism Spectrum Disorders in the Second Year of Life. *Journal of Autism and Developmental Disorders*, 34(5):473–493, October 2004. ISSN 0162-3257. doi: 10.1007/s10803-004-2544-y.

Jr. White K.P., T E Hutchinson, and J M Carley. Spatially dynamic calibration of an eye-tracking system. *Systems, Man and Cybernetics, IEEE Transactions on*, 23(4):1162–1168, July 1993.

JanM. Wiener, Christoph Holscher, Simon Büchner, and Lars Konieczny. Gaze behaviour during space perception and spatial decision making. *Psychological Research*, 76(6):713–729, 2012.

Oliver Williams, Andrew Blake, and Roberto Cipolla. Sparse and semi-supervised visual mapping with the S3GP. In *Computer Vision and Pattern Recognition*, pages 230–237, 2006.

Hugh R Wilson, Frances Wilkinson, Li-Ming Lin, and Maja Castillo. Perception of head orientation. *Vision Research*, 40(5):459–472, March 2000.

William Hyde Wollaston. On the Apparent Direction of Eyes in a Portrait. *Philosophical Transactions of the Royal Society of London*, 114:pp. 247–256, 1824.

Junwen Wu and Mohan M Trivedi. A Two-stage Head Pose Estimation Framework and Evaluation. *Pattern Recogn.*, 41(3):1138–1158, March 2008.

Ying Wu and K Toyama. Wide-range, person- and illumination-insensitive head orientation estimation. In *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pages 183–188, 2000.

J Xiao, T Moriyama, T Kanade, and J Cohn. Robust Full-Motion Recovery of Head by Dynamic Templates and Re-registration Techniques. 13(1):85–94, 2003.

J Xiao, S Baker, and I Matthews. Real-Time Combined 2D+3D Active Appearance Models. In *IEEE Conf. Comp. Vision & Pattern Recognition (CVPR)*, 2004a.

Jing Xiao, T Kanade, and J F Cohn. Robust full-motion recovery of head by dynamic templates and re-registration techniques. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pages 156–162, May 2002.

Jing Xiao, Simon Baker, Iain Matthews, and Takeo Kanade. Real-Time Combined 2D+3D Active Appearance Models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 535–542, June 2004b.

Xuehan Xiong and Fernando De la Torre Frade. Supervised Descent Method and its Applications to Face Alignment. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, May 2013.

Xuehan Xiong and Fernando De la Torre. Supervised Descent Method for Solving Nonlinear Least Squares Problems in Computer Vision. *CoRR*, abs/1405.0, 2014.

Xuehan Xiong, Zicheng Liu, Qin Cai, and Zhengyou Zhang. Eye Gaze Tracking Using an RGBD Camera: A Comparison with a RGB Solution. In *International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, pages 1113–1121, 2014.

Hirotake Yamazoe, Akira Utsumi, Tomoko Yonezawa, and Shinji Abe. Remote gaze estimation with a single camera based on facial-feature tracking without special calibration actions. In *Symposium on Eye tracking Research & Applications*, volume 1, pages 245–250. ACM, 2008.

Shuicheng Yan, Zhenqiu Zhang, Yun Fu, Yuxiao Hu, Jilin Tu, and Thomas Huang. Learning a Person-Independent Representation for Precise 3D Pose Estimation. In Rainer Stiefelhagen, Rachel Bowers, and Jonathan Fiscus, editors, *Multimodal Technologies for Perception of Humans SE - 28*, volume 4625 of *Lecture Notes in Computer Science*, pages 297–306. Springer Berlin Heidelberg, 2008.

Ruigang Yang and Zhengyou Zhang. Model-based head pose tracking with stereovision. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pages 255–260, May 2002.

P Yao, G Evans, and A Calway. Using affine correspondence to estimate 3-D facial pose. In *International Conference on Image Processing*, volume 3, pages 919–922 vol.3, 2001.

Dong Hyun Yoo and Myung Jin Chung. A novel non-intrusive eye gaze estimation using cross-ratio under large head motion. *Computer Vision and Image Understanding*, 98(1): 25–51, April 2005.

Alan L Yuille, Peter W Hallinan, and David S Cohen. Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, 8(2):99–111, August 1992.

W Zhang, Q Wang, and X Tang. Real Time Feature Based 3D Deformable Face Tracking. In *Proc. of European Conference on Computer Vision, Marseille*, 2008.

Gangqiang Zhao, Ling Chen, Jie Song, and Gencai Chen. Large head movement tracking using sift-based registration. In *Proceedings of the 15th international conference on Multimedia*, MULTIMEDIA '07, pages 807–810, New York, NY, USA, 2007. ACM.

Qi Zhao and Christof Koch. Learning saliency-based visual attention: A review. *Signal Processing*, 93(6):1401–1407, June 2013.

Mingcai Zhou, Lin Liang, Jian Sun, and Yangsheng Wang. AAM based face tracking with temporal matching and face segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 701–708, June 2010.

Xiangxin Zhu and Deva Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Computer Vision and Pattern Recognition*, pages 2879–2886. IEEE Computer Society, June 2012. ISBN 978-1-4673-1226-4.

Zhiwei Zhu, Qiang Ji, and K P Bennett. Nonlinear Eye Gaze Mapping Function Estimation via Support Vector Regression. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 1, pages 1132–1135, 2006.

Kenneth Alberto Funes Mora
08^{th} October 1985, Costarrican
Lausanne, Switzerland
kenneth.funes@idiap.ch
www.idiap.ch/~kfunes

EDUCATION

PhD Student in Electrical Engineering. École Polytechnique Fédérale de Lausanne. Since 2011.

Master of Science in Computer Vision and Robotics. Erasmus Mundus ViBot. Université de Bourgogne (France), Universitat de Girona (Spain) and Heriot-Watt University (United Kingdom). Master Thesis at INRIA Rhône-Alpes. 2008-2010.

License in Electronics Engineering (5 years). Costa Rica Institute of Technology, ITCR. Accredited by the Canadian Engineering Accreditation Board (CEAB). 2003-2008.

RESEARCH INTERESTS

- 3D face tracking and gaze modeling (PhD thesis)
- Computer vision, machine learning and statistical modeling
- Human behavior and interaction analysis from automated methods
- Multimodal data processing
- Computer graphics
- Digital design and reconfigurable systems

EMPLOYMENT

Research Assistant, Idiap Research Institute, Martigny, Switzerland. January 2011-present.

Research intern, LEAR Team, INRIA Rhône-Alpes. Grenoble, France (January to August 2010).

Research intern, Computer Vision Laboratory. University of Girona, Spain (Summer 2009).

Research and development engineer, Canam Technology Inc. 2007-2008

Student assistant, SIP-Lab, Costa Rica Institute of Technology. 2005-2007

MAIN PROJECTS

3D face tracking and gaze modeling: PhD thesis project developed at the Idiap Research Institute under the supervision of Dr. Jean-Marc Odobez:
We have developed diverse techniques for 3D gaze tracking using consumer RGB-D sensors. We address the challenges of low-resolution and appearance variations due to head pose, user and sensing conditions. We also address higher-level problems in human-robot interaction, human computer interaction, sociology and psychology studies. Developed within the SNFS projects: TRACOME, G3E, SONVB and UBImpressed.

Robust face descriptors in uncontrolled settings: MSc thesis at the LEAR Team, INRIA Rhône-Alpes under the supervision of Matthieu Guillaumin, Jakob Verbeek and Cordelia Schmid:
We studied different variants of facial image descriptors for the task of face verification in unconstrained scenarios. The variants included different image alignment methodologies in combination with local and/or holistic descriptors.

PASCAL VOC 2009: We worked on the implementation of object classification algorithms from natural images. Computer Vision Laboratory, University of Girona.

Digital audio broadcasting (DAB) transmitter. Canam Technology Inc. I developed a DAB transmitter from reconfigurable hardware (FPGAs) and high-level software for transmission control. This transmitter was used in a DAB rebroadcasting system.

CONFERENCE PUBLICATIONS (PEER REVIEWED)

- **Who Will Get the Grant ? A Multimodal Corpus for the Analysis of Conversational Behaviours in Group**. Catharine Oertel, *Kenneth Alberto Funes Mora*, Samira Sheikhi, Jean-Marc Odobez and Joakim Gustafson. in: International Conference on Multimodal Interaction, Understanding and Modeling Multiparty, Multimodal Interactions Workshop, Istanbul, Turkey, 2014

- **Geometric Generative Gaze Estimation (G^3E) for Remote RGB-D Cameras**. Kenneth Alberto Funes Mora and Jean-Marc Odobez. in: IEEE Computer Vision and Pattern Recognition Conference, Columbus, Ohio, USA. June 2014

- **3D Gaze Tracking and Automatic Gaze Coding from RGB-D Cameras**. Kenneth Alberto Funes Mora and Jean-Marc Odobez. in: IEEE Conference in Computer Vision and Pattern Recognition, Vision Meets Cognition Workshop, Columbus, Ohio, USA, 2014

- **EYEDIAP: A Database for the Development and Evaluation of Gaze Estimation Algorithms from RGB and RGB-D Cameras**. Kenneth Alberto Funes Mora, Florent Monay and Jean-Marc Odobez. in: ACM Symposium on Eye Tracking Research and Applications, Florida, United States of America. March 2014

- **A Semi-Automated System for Accurate Gaze Coding in Natural Dyadic Interactions**. Kenneth Alberto Funes Mora, Laurent Son Nguyen, Daniel Gatica-Perez and Jean-Marc Odobez in: ACM International Conference on Multimodal Interaction, Sydney, Australia. December 2013

- **3D Head Pose and Gaze Tracking and Their Application to Diverse Multimodal Tasks**. Kenneth Alberto Funes Mora in: *Doctoral Consortium* of the ACM International Conference on Multimodal Interaction, Sydney, Australia. December 2013

- **Person Independent 3D Gaze Estimation From Remote RGB-D Cameras**. Kenneth Alberto Funes Mora and Jean-Marc Odobez in: IEEE International Conference on Image Processing, Melbourne, Australia. September 2013

- **Gaze Estimation from Multimodal Kinect Data**. Kenneth Alberto Funes Mora and Jean-Marc Odobez in IEEE CVPR Workshop on Gesture Recognition. June 2012. Providence, United States. [Oral presentation, *Best student paper award*]

JOURNAL PUBLICATIONS

- **Gaze Estimation in the 3D Space Using RGB-D sensors. Towards Head-Pose And User Invariance**. Kenneth Alberto Funes Mora and Jean-Marc Odobez. **(under review)**

OTHER PUBLICATIONS

- **A Gaze Estimation Method and Apparatus**. Kenneth Alberto Funes Mora and Jean-Marc Odobez. PCT EP Patent application PCT/EP2014/062604.

- **EYEDIAP Database: Data Description and Gaze Tracking Evaluation Benchmarks**. Kenneth Alberto Funes Mora, Florent Monay and Jean-Marc Odobez. Idiap Tech Report. Idiap-RR-08-2014. May 2014.

SUMMER SCHOOLS, WORKSHOPS, TALKS AND CHALLENGES

- "3D head pose and gaze tracking using remote RGB-D sensors". Invited talk at the *KTH Speech, music and hearing group*. Stockholm, Sweden. April 2014
- "A model for person-independent gaze estimation from RGB-D cameras". Talk at the *Machine Learning Workshop*. École Polytechnique Fédérale de Lausanne. November 2012
- "Sensing and Analyzing Nonverbal Behavior in Interactions at Work". Talk at the *SONVB Workshop*. Neuchâtel, Switzerland. September 2012

- Participated in the *International Create Challenge*. Bulb-me-back project: start-up proposal and prototype of a question forwarding website. Idiap Research Institute. September 2012.
- Attended the INRIA Visual Recognition and Machine Learning Summer School. Grenoble, France. August 2010.

RECOGNITIONS AND GRANTS

- **Idiap PhD Student Research Award**: Idiap Research Institute 2014. Martigny, Switzerland.
- **Travel grant**: Doctoral consortium at ICMI 2013. Sydney, Australia.
- **Best student paper award**. CVPR 2012 Workshop on Gesture Recognition. June 2012.
- **Erasmus Mundus grant:** awarded by the European commission to pursue the M.Sc. studies.
- M.Sc. Mention **trés bien** (Université de Bourgogne) and with **distinction** (Heriot-Watt University).
- **Honor grant** for most of my undergraduate studies.

SERVICE

- Reviewer: TCSVT; MVAP; VAAM; CVIU; THMS and ACCMM.
- Member of the student committe at the Electrical Engineering department at EPFL [May 2012-April 2014].
- Student supervision: Matthieu Duval (MSc thesis, 6 months, 2013).

SKILLS

General

Knowledge in Pattern Recognition, Computer Vision, Machine Learning, Image Processing, Medical Imaging and Electronic Engineering design (Digital Systems, Electric Communications, Power Electronics, Automatic Control and Telecommunications)

Computer related

Programming languages: Python, C, C++, C#, Java, Verilog HDL, Assembler, PICC, MATLAB®.

Libraries and tools: OpenCV, OpenGL, Point Cloud Library, CUDA, OpenCL, OpenKinect, OpenNI, Qt, Doxygen, SWIG, Python API, Make, CMake, subversion, Git, Inkscape, Gimp.

Web development: CSS, HTML, Django, PHP (basic).

Operative Systems: Mac OS X, Linux [various distributions], Windows.

LANGUAGES

Spanish [Mother tongue]

English [Written and spoken]

French [Intermediate].

HOBBIES

- Playing and writing music.
- Drawing and painting.
- Traveling and socializing.